

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO PODPORU MANAGEMENTU RIZIK V IT PROJEKTECH

DIPLOMOVÁ PRÁCE

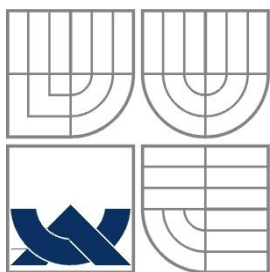
MASTER'S THESIS

AUTOR PRÁCE

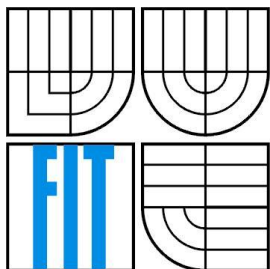
AUTHOR

Bc. KRISTIÁN BIRKUS

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO PODPORU MANAGEMENTU RIZIK V IT PROJEKTECH

SYSTEM FOR RISK MANAGEMENT SUPPORT IN IT PROJECTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. KRISTIÁN BIRKUS

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. ŠÁRKA KVĚTOŇOVÁ, Ph.D.

BRNO 2009

Zadání diplomové práce

Řešitel: **Birkus Kristián, Bc.**

Obor: Informační systémy

Téma: **Systém pro podporu managementu rizik v IT projektech**

Kategorie: Softwarové inženýrství

Pokyny:

1. Seznamte se s problematikou projektového řízení. Zaměřte se oblast managementu rizik v projektech, které jsou typické pro IT oblast.
2. Specifikujte požadavky na programovou podporu managementu rizik (možnosti definování procesů, vytváření/import/export vstupů/výstupů pro procesy řízení rizik, vytváření checklistů pro různé úrovně projektu, statistické souhrny výsledků, generování závěrečných zpráv atp.).
3. Navrhněte koncepci programové podpory managementu rizik v IT projektech.
4. Implementujte prototyp navrženého systému ve vhodně zvoleném prostředí.
5. Funkčnost ověřte na ukázkovém projektu a zhodnoťte dosažené výsledky se zaměřením na použitelnost v reálném prostředí.

Literatura:

- Rosenau, M.D.: Řízení projektů, Computer Press, 2003, 344 s. ISBN 80-7226-218-1
- Kliem, L. R.: Project Management Methodology. Marcel Dekker Inc., 1997.
- Schwalbe, K.: Řízení projektů v IT - Kompletní průvodce, Computer Press, 2007, ISBN: 978-80-251-1526-8
- Schulte, P.: Complex IT Project Management. AUERBACH PUBLICATION, 2004, ISBN 0849319323

Při obhajobě semestrální části diplomového projektu je požadováno:

- Body 1 - 3.

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci ročníkového a semestrálního projektu (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Květoňová Šárka, Ing., UIFS FIT VUT**

Datum zadání: 22. září 2008

Datum odevzdání: 26. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Táto diplomová práca sa zaoberá systémom pro podporu managementu rizik v IT projektech. Byla potrebná dôkladná analýza problematiky projektového řízení a managementu rizik zaměřená na oblasti typické pro IT. Po důkladné analýze bylo možné implementovat samotný systém. Systém byl implementován v programovacím jazyce C#, a na databázové úrovni pracuje s databázovým serverem MS SQL.

Abstract

This thesis presents a system designed to support risk management in IT projects. The presented system is based on an in depth analysis of project and risk management in the field of information Technologies. The implementation started only after the exhaustive analyzation process. The system was implemented in programming language C#. On the database level MS SQL server is used.

Klíčová slova

projekt, projektový management, řízení projektu, trojimperatív, procesy, riziko, management rizik, analýza rizik, identifikace rizik, softwarová podpora

Keywords

project, project management, processes, risk, risk management, risk analysis, risk identification, software support

Citace

Birkus Kristián: Systém pro podporu managementu rizik v IT projektech, diplomová práce, Brno, FIT VUT v Brně, 2009

Systém pro podporu managementu rizik v IT projektech

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Šárky Květoňové, Ph.D. Uvedl jsem všechny literární prameny a publikace, z kterých jsem čerpal.

.....

Kristián Birkus

12.5.2009

Poděkování

Na tomto místě bych chtěl poděkovat mé vedoucí Ing. Šárce Květoňové, Ph.D. za odborné vedení, cenné rady, konzultace a připomínky, které přispěli k dokončení diplomové práce.

© Kristián Birkus, 2009

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Projekt a projektový manažment.....	4
2.1 Projekt.....	4
2.1.1 Definícia projektu.....	4
2.1.2 Trojimperatív.....	6
2.1.3 Ciele projektu.....	8
2.1.4 Životný cyklus projektu.....	8
2.1.5 Prostredie projektu.....	10
2.2 Projektový manažment.....	11
2.2.1 Projektové riadenie a riadenie projektov.....	11
2.2.2 Formy projektového manažmentu.....	12
2.2.3 Predpoklady úspechu projektového manažmentu.....	12
3 Procesy riadenia projektov.....	14
3.1 Overenie realizovateľnosti projektov.....	14
3.2 Riadenie času.....	15
3.3 Riadenie kvality.....	15
3.4 Riadenie nákladov.....	16
3.5 Riadenie zmeny.....	16
4 Riadenie rizík v IT projektoch.....	17
4.1 Definícia pojmu riziko.....	17
4.2 Význam riadenia rizík v projektoch.....	17
4.3 Plánovanie riadenia rizík.....	18
4.4 Identifikácie rizík.....	20
4.4.1 Námety pre identifikáciu rizík.....	20
4.4.2 Register rizík.....	22
4.5 Kvalitatívna analýza rizík.....	22
4.5.1 Výpočet rizikových faktorov pomocou matice pravdepodobnosti a dôsledkov.....	22
4.5.2 Sledovanie desiatich najrizikovejších položiek.....	24
4.5.3 Expertné posudzovanie.....	24
4.6 Kvantitatívna analýza rizík.....	24
4.6.1 Rozhodovacie stromy a očakávaná peňažná hodnota.....	25
4.6.2 Simulácia.....	25
4.6.3 Analýza citlivosti.....	26
4.7 Plánované reakcie na riziká.....	26
4.8 Sledovanie a kontrola rizík.....	27
4.9 Softwarová podpora riadenia rizík v projektoch.....	27
5 Analýza a návrh.....	28
5.1 Neformálna špecifikácia požiadaviek.....	28
5.1.1 Očakávané funkcie systému.....	29
5.2 Diagram prípadu použitia.....	31
5.3 Návrh architektúry.....	32
5.4 Návrh databázy.....	33

5.5	Návrh grafického rozhrania.....	35
6	Implementácia.....	37
6.1	Implementačné nástroje	37
6.1.1	.NET Framework.....	37
6.1.2	Jazyk C#.....	38
6.1.3	MS SQL Server	38
6.1.4	Microsoft Visual Studio 2005	38
6.1.5	ADO .NET	39
6.2	Implementácia databázy.....	39
6.2.1	Vytvorenie databázových tabuliek	39
6.2.2	Prístup k dátovej vrstve.....	40
6.3	Hlavné funkcie systému	42
6.3.1	Prihlásenie a odhlásenie zo systému.....	42
6.3.2	Pridávanie a správa používateľov.....	43
6.3.3	Výber a otvorenie projektov.....	44
6.3.4	Výpočet očakávanej peňažnej hodnoty.....	45
6.3.5	Vytvorenie nového projektu.....	46
6.3.6	Rozhranie projektu a jeho rizík	47
6.3.7	Správa projektu.....	48
6.3.8	Správa procesov projektu.....	49
6.3.9	Správa rizík	50
6.3.10	Priradenie súborov k rizikám	52
6.3.11	Kvalitatívne hodnotenie a časový graf.....	53
6.3.12	Mapa rizík	55
6.3.13	Export / Import rizík	57
6.3.14	Karta rizika.....	58
6.3.15	Zmena vlastných údajov používateľa	59
6.3.16	Správa ďalších údajov týkajúce sa rizík.....	60
7	Testovanie.....	61
8	Ďalší rozvoj systému	62
9	Záver	63

1 Úvod

V súčasnosti existuje mnoho teoreticky opísaných postupov ako riešiť rôzne typy projektov, však napriek tomu neustále v praxi vidíme ako niektoré projekty končia neúspechom. Jedným problémom pri riešení projektu je neschopnosť projektových manažérov preniesť teóriu do praxe. Vzhľadom na to, že teória dokonale opisuje ako úspešne riešiť projekty, je percento neúspešných projektov vysoké.

Úspech alebo neúspech projektu nie je závislý na jedinom rozhodnutí, na jedinom kroku v priebehu projektu, ale závisí od mnohých faktorov. Ak však nebudeme dbať na znalosti získané z teórie a na skúsenosti z praxe, môžeme nevedomky spraviť mnoho chýb, ktoré sa podpíšu na zlyhaní projektu. Teória nie sú len prázdne slová, ale je to návod ako robiť veci lepšie a chytrejšie. Nemali by sme preto brať teoretické znalosti na ľahkú váhu a mali by sme sa v čo najväčšej miere usilovať preniesť teóriu do praxe.

Cieľom tejto diplomovej práce je navrhnuť a implementovať taký softwarový systém, ktorý pomôže tieto neúspechy v oblasti manažmentu zdokonaľiť. Je zameraná na podporu rizík v IT projektoch, lebo v tejto oblasti celého procesu projektového riadenia sú najväčšie nedostatky. A tak práve manažment rizík je v procese riadenia pre mnohé projekty osudným. Veľká časť projektov, ktorá manažmentu rizík nevenovala pozornosť, skončila neúspechom [6].

Druhá kapitola práce je venovaná popisu projektu a projektového manažmentu. Sú tu popísané základné pojmy týkajúce sa projektu ako definície, ciele, prostredia a zúčastnené osoby projektov. Kapitola obsahuje aj popis samotného projektového manažmentu, vznik a podstaty. Vymenované sú aj jednotlivé formy projektového manažmentu a predpoklady úspechu pri riadení.

Tretia kapitola sa týka procesov riadenia a overenia realizovateľnosti projektu. Sú tu krátko zmienené jednotlivé procesy ako riadenie času, nákladov, kvality a zmien.

Štvrtá kapitola sa zaoberá už samotným manažmentom rizík. Sú tu popísané definície pojmov, význam riadenia rizík a jednotlivé metódy riadenia. U jednotlivých metód sú znázornené ich použitia v riadeniach IT projektoch.

Piata kapitola popisuje samotnú analýzu a návrh systému. Na začiatku kapitoly sú definované neformálne špecifikácie a očakávané funkcie systému. Tieto funkcie sa delia na funkcie týkajúce sa používateľov, projektov, procesov a rizík. Kapitola obsahuje aj diagram prípadu použitia definujúci chovanie systému z pohľadu zákazníka. Z týchto neformálnych špecifikácií vznikol formálny návrh systému, ktorý je rozdelený na návrh architektúry, databázy a grafického rozhrania. V návrhu databázy je uvedený aj databázový model zobrazený pomocou ER diagramu.

Na začiatku šiestej kapitoly sú uvedené implementačné prostriedky, ktoré som používal pri samotnom vývoji. Kapitola pokračuje s vytvorením databázových tabuliek a možnosťou prístupu k dátam obsadených v týchto tabuľkách. Ako posledné sú popísané všetky hlavné funkcie systému ktoré boli implementované podľa formálneho návrhu. Niektoré funkcie sú demonštrované obrázkami.

Siedma kapitola sa zaoberá testovaním systému, postupmi testovania a s najväčšími chybami, ktoré boli zistené pri tejto fáze vývoja.

Do ôsmej kapitoly som zhrnul všetky nápady o možných rozšíreniach, ktoré ma napadli pri implementácii a testovaní systému. Tieto rozšírenia by mohli byť primárnymi funkciami, pri budúcom zdokonaľovaní aplikácie.

2 Projekt a projektový manažment

Projektový manažment je nový prístup k riadeniu projektov s presne vymedzeným cieľom, ktorý musí byť dosiahnutý v stanovenom čase, nákladoch a kvalite. Projekt v hospodárskej praxi je pomerne zložitá, nová a dynamická úloha, ktorú je potrebné splniť. V hodnotovom systéme má veľký význam a rieši sa v jedinečných situačných podmienkach. Úspešné riešenie projektových úloh predpokladá uplatnenie množstva metód, techník postupov a nástrojov hlavne v plánovaní, organizovaní a kontrole projektových prác.

2.1 Projekt

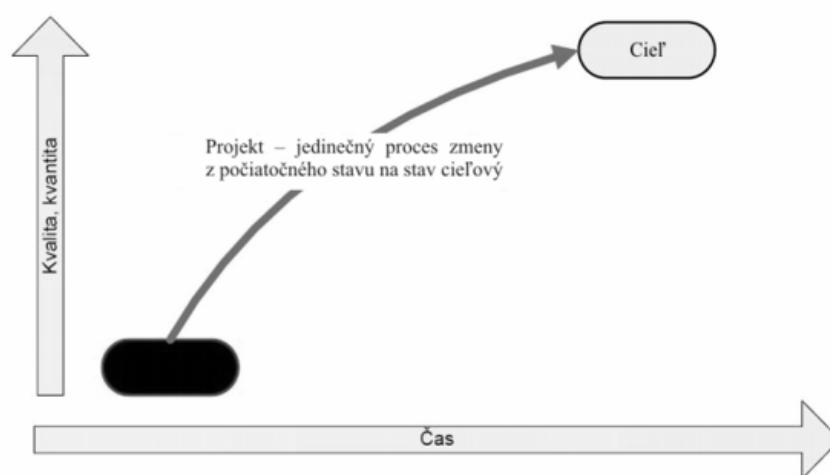
Najdôležitejším pojmom projektového manažmentu je projekt. Tento pojem pochádza z anglosaského poňatia slova „Project“, ktoré označuje proces plánovania a riadenia rozsiahlych operácií. Neexistuje jedna obecná definícia projektu, pričom každá organizácia pod týmto pojmom chápe niečo iné.[9]

2.1.1 Definícia projektu

Projekt je jedinečný proces, ktorý sa skladá zo súboru koordinovaných a riadených činností s dátumom začatia a dokončenia, vykonávaných na dosiahnutie cieľa zodpovedajúceho špecifickým požiadavkám vrátane časových, nákladových a zdrojových obmedzení.

Projekt je v čase konečný súbor technologicky a organizačne nadväzujúcich činností a prostriedkov potrebných na vytvorenie výrobku, služby či inej zmeny, užitočných pre určitého zákazníka.

Ako je vidieť na obrázku 2.1 projekt je úsilím, v ktorom sú ľudské, materiálne a finančné zdroje organizované špecifickým spôsobom za účelom vykonania súboru konkrétnych činností v rámci vymedzených nákladov a času, ktoré vedú k dosiahnutiu určeného cieľa [7].



Obrázok 2.1: Definícia projektu [7]

Pre všetky projekty existuje veľa spoločných rysů ale **projekt, ako ho dnes chápeme** je charakteristický najmä [7]:

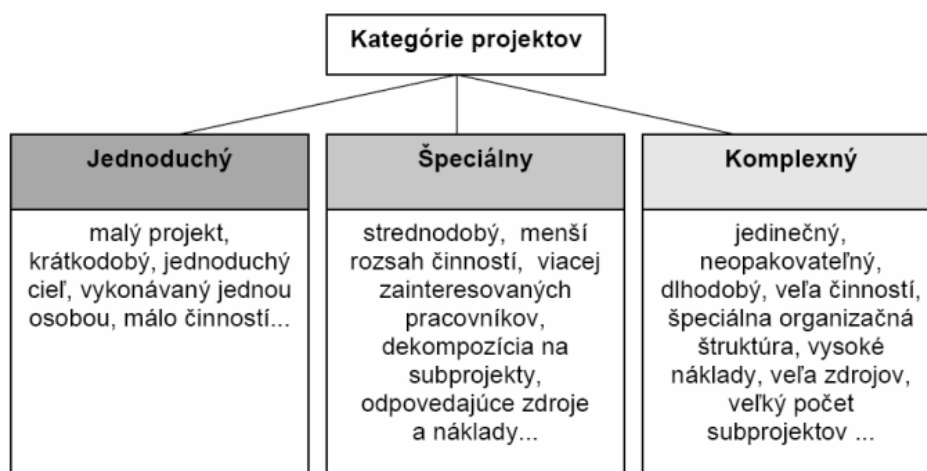
- jedinečnosťou a unikátnosťou súboru činností, ktoré sa odlišujú od rutinných činností nielen svojím obsahom, ale aj cieľovým zameraním,

- definovaním stratégie, ktorá vedie k dosiahnutiu stanovených cieľov,
- potrebnými zdrojmi a nákladmi,
- presne vymedzeným časom, stanovením jeho začiatku a konca,
- požadovanou kvalitou výstupu.

Medzi ďalšie charakteristiky projektu patrí:

- jedinečná organizácia štruktúry a procesov projektu,
- premenlivá interdisciplinárna spolupráca zvyčajne nezohratých partnerov,
- istá miera rizika a neistoty a podobne.

Ani pri hľadaní kategórií projektov nenájdeme jednotné hodnotenie. Projekty delia z hľadiska zamerania na hospodárske a ostatné a z hľadiska zložitosti na jednoduché a zložité. Odlišné **rozdelenie projektov** podľa obsahu uvádza [2], kde sú projekty delené na jednoduché, špeciálne a komplexné. Toto rozdelenie je znázornené na obrázku 2.2.



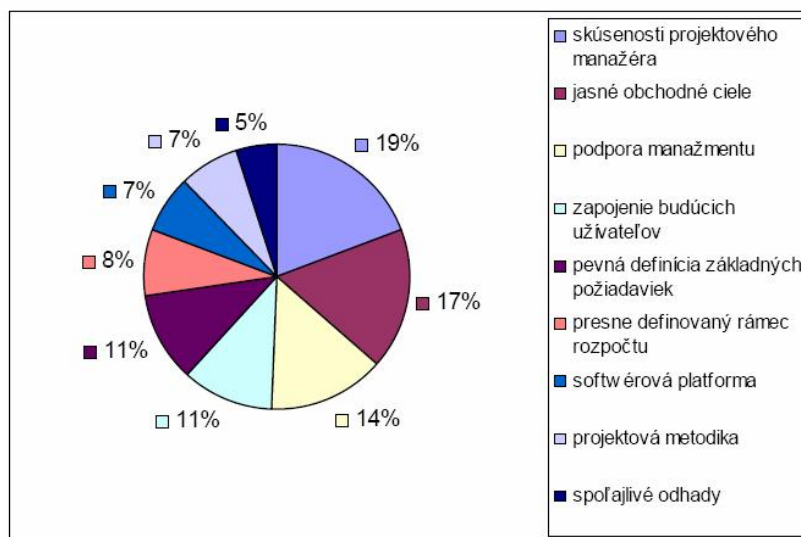
Obrázok 2.2: Kategória projektov podľa obsahu [8]

Úspešné riešenie projektových úloh predpokladá uplatnenie množstva metód, techník, postupov a nástrojov hlavne v plánovaní, organizovaní a kontrole projektových prác [9].

Za úspešný projekt považujeme taký, ktorý je ukončený:

- v stanovenom termíne a nie je prekročený rozpočet,
- je splnení rozsah a špecifikácia projektu v požadovanej kvalite, tzn. je splnená akceptácia výsledku zákazníkom,
- s minimálnym množstvom alebo vzájomne odsúhlasenými zmenami, čo sa týka zadania projektu,
- bez toho, aby bola narušená plynulosť hlavnej práce v organizácii.

Graf ktorý je znázornený na obrázku 2.3, zhrňuje hlavné faktory ovplyvňujúce úspešnosť projektu vyplývajúce zo štúdie pre úspešnosti projektov realizovaných spoločnosťou Logos v roku 2007 [9].



Obrázok 2.3: Hlavné faktory ovplyvňujúce úspešnosť projektu [9]

Podľa prieskumu týkajúceho sa oblasti projektového manažmentu realizovanou v Českej republike na jar minulého roku firmou Ernst&Young, sa pri uvádzaní príčin zlyhania projektu, označujú päť hlavných dôvodov neúspechu projektov [9]:

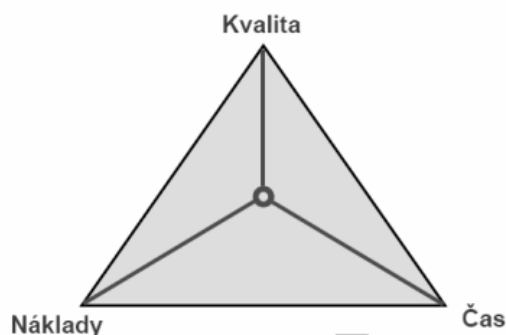
- zmena rozsahu projektu, ako dôsledok nesprávneho vymedzenia projektu,
- nesprávne personálne zabezpečenie projektov a koordinácia tímu,
- zmena rozsahu projektu v dôsledku následných zmien okolitého prostredia,
- rozdielne očakávania výstupov projektov,
- nedostatočný alebo príliš optimistický rozpočet/plán.

Pri výbere faktorov úspechu projektu sa najčastejšie používa, tzv. „magický trojuholník“, označovaný aj ako „trojimperatív“, kde jeho tri strany reprezentujú kvalita, náklady a termíny.

2.1.2 Trojimperatív

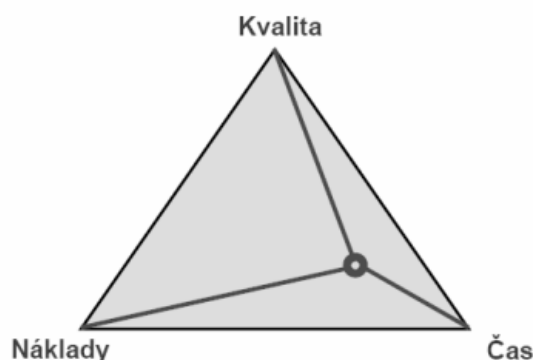
Trojimperatív je pojem, používaný v projektovom riadení na označenie troch navzájom pevne spätých prvkov, ktorými je projekt definovaný (obrázok 2.4). Jedná sa o:

- kvalitu (definovaná špecifikáciou prevedenia),
- čas (definovaný časovým plánom),
- náklady (definované dostupnosťou zdrojov – finančných, materiálových, ľudských) [8].



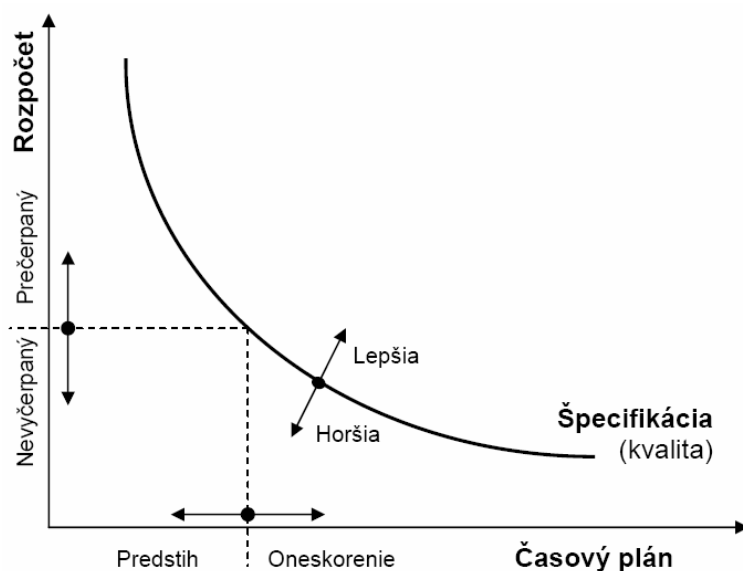
Obrázok 2.4: Tri prvky trojimperatívu [8]

Od úspešného projektu sa očakáva splnenie všetkých troch prvkov (cieľov) trojimperatívu, dokončenie projektu v stanovenom termíne, v požadovanej kvalite a v rozmedzí naplánovaných nákladov. Pri rôznych projektoch môžu mať jednotlivé prvky trojimperatívu rôznu váhu dôležitosti. Napríklad pre získanie konkurenčnej výhody na trhu je pre zadávateľa projektu dôležitejší čas a kvalita ako náklady. Zmena v jednom prvku trojimperatívu má vždy vplyv na zvyšné dva prvky (obrázok 2.5).



Obrázok 2.5: Odchýlenie sa trojimperatívu od štandardného postavenia [8]

Ak je napríklad potrebné projekt urýchliť, je možné alebo znížiť nároky na kvalitu výstupu projektu, alebo vyčleniť vyššie náklady (zvyšiť rozpočet, pridať ďalšie zdroje, povoliť nadčasovú prácu, objednať externé služby a pod.). Graf ktorý je zobrazený na obrázku 2.6 zachycuje práve toto rozdelenie, ako sú závislé tieto tri prvky riadenia. Ak je potrebné pri riešení projektu (napríklad výskumného či vývojového) dosiahnuť vyššiu kvalitu oproti schválenej špecifikácii, je možné to dosiahnuť predĺžením času riešenia projektu, alebo navýšením zdrojov (finančných, materiálových, ľudských) [8].



Obrázok 2.6: Vzájomná závislosť jednotlivých prvkov trojimperatívu [8]

Akúkoľvek zmenu trojimperatívu, ak je nevyhnutná, je potrebné konzultovať so zadávateľom projektu s cieľom ujasnenia si možností zmeny a priority prvkov trojimperatívu. Ak zmena ovplyvní zvyšné prvky trojimperatívu je potrebné požiadať o zmenu rozpočtu, časového plánu alebo špecifikácie. Každá zmena, ktorá je realizovaná prostredníctvom projektu, musí mať teda určenú jasnú štruktúru vzájomne previazaných cieľov, ktoré zachytávajú všetky podstatné súvislosti s uskutočňovanou zmenou [5].

Mnoho autorov ako je uvedené aj v literatúre [4] rozširujú trojimperatív aj o **d ďalšie faktory, ktoré súvisia so spôsobom, akým bude projekt prevedený.**

2.1.3 Ciele projektu

Predstavujú významné výsledky realizácie projektu, ku ktorým smeruje realizácia jednotlivých projektových činností. Je ich potrebné jasne špecifikovať na začiatku projektových prác, pretože bez jasne stanovených cieľov sa riadenie projektu stáva len náhodným procesom. Pri určovaní cieľov projektu by mal manažér projektu spolupracovať s projektovým tímom. Vždy je, ale nevyhnutné, aby ciele boli schválené zadávateľom projektu. Tak je možné predísť prípadným nedorozumeniam a neplánovaným zásahom v priebehu realizácie projektu.

Pri formulovaní cieľov projektu a stratégií vedúcich k ich dosiahnutiu by mali byť rešpektované nasledovné zásady [9]:

1. Ciele projektu je nutné formulovať tak, aby bolo možné priebežne jednoznačne kontrolovať priebeh ich plnenia a na záver vyhodnotiť mieru ich dosiahnutia. Nejasná formulácia cieľov silne ohrozuje priebeh a náklady projektu.
2. Pokiaľ nie sú jasne stanovené a dobre známe ciele projektu, nie je možné posudzovať, ktorý variant riešenia problému je optimálny.
3. Pri formulácii cieľov projektu je nutné vždy zvažovať, či je ich dosiahnutie reálne.
4. Cieľový stav projektu, ktorý má byť dosiahnutý, je treba chápať ako štruktúrovaný systém čiastkových cieľov.
5. Pri formulovaní cieľov projektu je nutné zvažovať všetky dôsledky realizácie projektu, ako aj pozitívne, tak i negatívne.

Projekty môžu mať rôznorodý charakter. Líšia sa predmetom riešenia, stupňom inovácie, rozsahom nákladov a časom potrebným na ich realizáciu. Dôležitým faktorom pri klasifikácii projektov je stupeň inovácie, ktorý nám hovorí, aké veľké zmeny projekt rieši. Ďalším faktormi kategorizácie projektov druh výstupu ich riešenia.

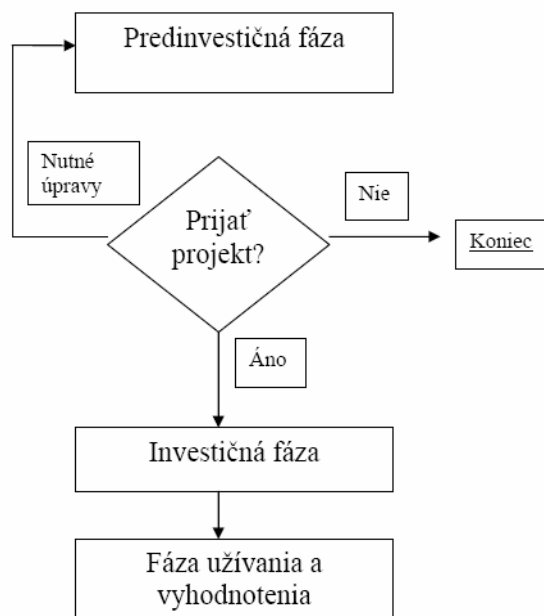
2.1.4 Životný cyklus projektu

Projekty sú dynamické systémy, ktoré sa i napriek svojej jedinečnosti a neopakovateľnosti, bez ohľadu na ich rozsah a druh, vyznačujú spoločným základnými postupmi, pozostávajúcimi z plánovania a riadenia procesov a spoločným **životným cyklom**.

Riadenie životného cyklu projektu znamená kontrolu, regulovanie a dohľad nad rôznymi aktivitami vykonávanými v jednotlivých fázach cyklu. **Všetky projekty bez ohľadu na rozsah a druh sa vyznačujú spoločnými základnými postupmi a životným cyklom, ktoré musia byť uskutočnené v rámci vymedzenej doby.** Vždy je však treba, aby za realizáciou projektu a kvalitu požadovaných výsledkov zodpovedala jedna osoba. Na záver sa uskutoční zhodnotenie celého projektu a definujú sa východiská, ktoré prispejú k eliminácii chýb a k níženiu rizikových udalostí pri realizácii projektu.

Ak chápeme riadenie projektov ako životný cyklus, potom ho môžeme rozdeliť na nasledujúce **tri fázy životného cyklu projektu** (Obrázok 2.7) :

- predinvestičná fáza (predbežné plánovanie a prípravu projektového zámeru),
- investičná fáza (podrobné plánovanie a realizácia projektu),
- fáza užívania a vyhodnotenia (tiež označovaná ako operačná fáza).



Obrázok 2.7: Fázy životného cyklu projektu [8]

Predinvestičná fáza je najdôležitejšia časť z celého projektu. Zahrňuje predbežné plánovanie a prípravu projektu. V úvodnej fáze je nutné stanoviť ciele a stratégiu projektu vedúcu k dosiahnutiu stanovených cieľov [8].

V investičnej fáze projektu je menovaný hlavný manažér projektu, ktorý je zodpovedný za riadenie projektu a projektový tím. Ďalej sa spracúvajú podrobné implementačné plány, definuje sa projektová organizácia, podrobné časové parametre, zdroje, náklady, realizujú sa výberové riadenia, výber dodávateľov, spracúva sa podrobná projektová dokumentácia a definujú sa dôsledky a špecifické podmienky súvisiace s realizáciou daného projektu.

Vo fáze užívania a záverečného vyhodnotenia projektu sa výsledky projektu predávajú do užívania a uskutočňuje sa záverečné komplexné vyhodnotenie projektu [4].

S pojmom životný cyklus projektu úzko súvisí aj fáza projektu a míľniky. **Fáza projektu** je časový úsek projektu, ktorý je vecne oddelený od ostatných častí. Majú vytýčené hranice, uľahčujú kontrolovanie čiastkových výsledkov projektu a korigovanie ďalšieho postupu. V závislosti od typu odvetvia a od stanovených podmienok v podniku sa používajú rôzne modely fáz. **Míľniky (Milestones)** sú významné udalosti v projekte, znamenajúce často zmenu fáz, rozhodnutie o opakovaní predmetnej fázy alebo ešte predchádzajúcej, o ukončení projektu alebo začatí nasledujúcej fázy. V projektovej praxi sa tieto fázy často prekrývajú. Projektový manažér používa aj interné míľniky, ktoré nie sú pre zákazníka viditeľné, a uľahčujú mu kontrolu a riadenie podniku. Fázy vo svojom súhrne predstavujú životný cyklus projektu. Riadenie životného cyklu projektu znamená kontrolu, a dohľad nad rôznymi aktivitami vykonávanými v jednotlivých fázach [9].

2.1.5 Prostredie projektu

Každý projekt je realizovaný v určitom prostredí – okolí podniku. Medzi projektom a okolím existujú určité vzťahy, ktoré môžu byť negatívne, ale rovnako aj pozitívne.

Projekt existuje a je realizovaný v [9] :

- určitom kultúrnom a sociálnom prostredí, ktoré ovplyvňuje chovanie, zvyky a rozhodovanie osôb zúčastňujúcich sa na projekte,
- konkrétnej medzinárodnej a politickej situácii, ktorá odráža celý rad aspektov od zákonného prostredia až po režim a komunikáciu v určitých časových pásmach, obvyklé obdobia pre dovolenku alebo náboženské sviatky,
- určitom hospodárskom a trhovom prostredí, ktoré môže generovať celý rad impulzov v širokom spektre od neočakávaných konkurenčných útokov po vstup nových technológií na trh,
- špecifickom fyzickom okolí, kde určité obmedzenia môžu vyplývať z geografických alebo geologických podmienok, ochrany životného prostredia, výkyvov počasia a podobne.

Kvalita projektového manažmentu je aj pri používaní rozsiahlych metodík, štandardov a pravidiel úplne závislá na ľuďoch, ktorí sú jeho nositeľmi, nie iba na ich individuálnom výkone, ale na aktivitách celého projektového tímu a jeho snahe dosiahnuť vytýčené ciele. Aby bol manažment projektu maximálne efektívnym procesom, je potrebné vytvoriť prechodnú štruktúru rolí, popísať vzťahy medzi týmito rolami, rozdeliť rozhodovaciu autoritu tak, aby úkony riadenia mali svoje oporné body, riadiace a výkonové zložky a aby bola jasne rozdelená zodpovednosť za splnenie jednotlivých úloh, za ich syntézu a splnenie celkového cieľa projektu [4].

Pretože projekt je jedinečný proces, nastáva v jeho priebehu a existencií celá rada situácií, ktoré nebolo možné predvídať. Pre podporu celkovej úspešnosti projektu je veľmi dôležité rozloženie záujmov, authority a rozhodovacích schopností, ktoré sú mapované prostredníctvom popisu záujmových skupín projektu.

Záujmové skupiny, projektu triedia jednotlivých interných účastníkov projektu alebo jednotlivcov a skupiny vonkajšieho prostredia so vzťahom k projektu podľa rozloženia ich individuálnych alebo skupinových cieľov. Identifikácia záujmových skupín projektu je jedným z prvých úloh spojených s prípravou a plánovaním projektu. Záujmové skupiny predstavujú rovnako jednotlivé osoby alebo skupiny, ktoré majú rôznu úroveň zodpovednosti a rozhodovacej authority vzhľadom ku konkrétnemu projektu [5].

V rámci každého projektu je možné identifikovať nasledovné základné záujmové skupiny, ktoré majú záujem na realizácii projektu a sú s ním jednoznačne spojené [8]:

- predstavitelia zákazníka, resp. zadávateľa projektu,
- predstavitelia dodávateľa projektu,
- predstavitelia užívateľa projektu.

Každý projekt má svojho **zákazníka**, ktorý má záujem na realizácii projektu a je jeho investorom alebo zadávateľom. Jedná sa spravidla o budúceho **užívateľa** výstupu projektu alebo o investora, pre ktorého znamená realizácia projektu zvýšenie potenciálnej úspešnosti na trhu prostredníctvom nového produktu alebo služby, ktorá je predmetom projektu. Kľúčovou osobou v skupine zákazníka projektu je **sponzor** projektu, ktorý je formálnym nositeľom spravidla najvyššej rozhodovacej authority v projekte. Sponzor projektu je spravidla funkčným manažérom zákazníka projektu, t. j. organizačnej jednotky, ktorá bude výsledok projektu používať. **Dodávateľ** projektu je spoločnosť alebo jej časť, ktorá je priamym účastníkom kontraktu a z neho plynúcej zodpovednosti

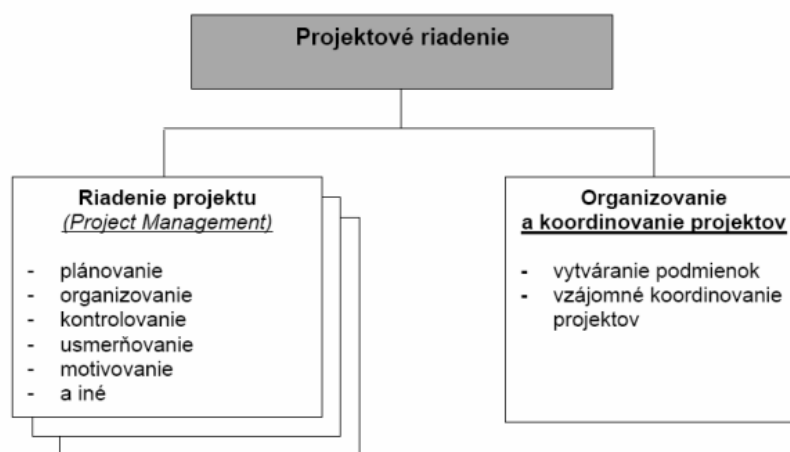
za vlastnú realizáciu projektu. Záujmom dodávateľa projektu je naplnenie podmienok kontraktu a získanie s tým spojenej odmeny [8].

2.2 Projektový manažment

Projektový manažment v zmysle manažérskej disciplíny existuje od vzniku našej civilizácie. Mnohé výdobytky starého sveta, ako napríklad plánovanie a stavba pyramíd v Gize, monumentálne egyptské chrámy, Čínsky múr, sedem divov starého sveta a pod., predstavujú úspešné projekty, ktoré museli mať riadené úlohy, ako náročné projekty dnes [9].

2.2.1 Projektové riadenie a riadenie projektov

Správne vymedzenie pojmu projektové riadenie (projektový manažment) je neľahké, pretože názory súčasných autorov sú niekedy až protichodné. Príčinu tohto stavu je možné hľadať v prekladoch anglického výrazu project management. Tento pojem je niekedy prekladaný do slovenčiny ako manažment projektu (riadenie projektu), no inokedy ako projektový manažment (projektové riadenie) vo význame riadenia organizácie pomocou projektov. Vzťah pojmov projektové riadenie a riadenie projektov je vyjadrený na upravenom obrázku 2.8 [8].



Obrázok 2.8: Vzťah projektového riadenia a riadenia projektu [8]

Implementácia (zavedenie) projektového riadenia do podniku (inštitúcie, firmy) pozostáva z analýzy, systematizácie a zlepšovania už používaných praktík projektového riadenia, ako aj z uplatňovania celkom nových praktík spojených s reorganizáciou projektovo – orientovanej organizácie.

Implementácia projektového riadenia je zvyčajne vykonávaná podľa nasledovných krokov [8]:

1. rozhodnutie vrcholového manažmentu o implementácii projektového riadenia do organizácie,
2. vypracovanie koncepcie implementácie projektového riadenia a sprievodného vzdelávania zúčastnených osôb,
3. realizácia vo vybranom pilotnom projekte,
4. vyhodnotenie výsledkov pilotného projektu a ich využitie pre ďalšie projekty.

Implementáciu projektového riadenia je možné považovať za projekt, ktorý má stanovený začiatok a koniec, definované disponibilné zdroje a požadovanú kvalitu výstupov. Pre vhodnú implementáciu zásad projektového riadenia do praxe je potrebné vytvoriť vhodnú organizačnú

štruktúru projektovej organizácie, ktorá rešpektuje špecifiká zakladajúcej organizácie a náročnosť daného projektu [8].

2.2.2 Formy projektového manažmentu

Väčšina projektov má vysoký stupeň technickej, marketingovej alebo výrobnnej zložitosti, ako aj vysoký stupeň inovačného alebo technického rizika a vyžaduje účasť niekoľkých separátnych organizácií – interných, externých, funkčných alebo iných. Aktuálne formy projektovej organizácie sa však uplatňujú rozlične, v závislosti od povahy. Podstaty projektu, organizácie a prostredia [12].

Projektový manažment je v zahraničnej literatúre nazývaný rozlične. Odhliadnuc od názvu všetky jeho formy charakterizujú dve črty (stránky). Tvorí ho [11]:

- projektový tím alebo projektová organizácia zostavená na riešenie,
- projektový manažér, ktorý zodpovedá za dosiahnutie cieľov.

Charakter oboch stránok závisí od aplikačnej oblasti. V americkej literatúre sa zvyčajne uvádzajú tieto formy projektového manažmentu [12]:

- Basic Projects Management (základný, bežný projektový manažment),
- Program Management (riadenie programov),
- New Venture Management (manažment nových podnikov),
- Product Management (výrobný manažment),
- Task force, resp. Ad hoc Management.

Basic Projects Management - je v podstate projektový manažment v európskom chápaní

Program Management – sa termínovo často používa ako synonymum projektového manažmentu, pretože program a projekt majú viacero spoločných črt. Sú to napríklad časové etapy, plnenie stanovených cieľov, plány rozpočtu a vlastnosti orientované na výstupy.

New Venture Management - je typ projektového manažmentu, ktorý sa vo firmách orientovaných na zákazníka., napr. pri generácii nových výrobkov alebo trhu. Tvorí sa špecializovaný tím na hľadanie nových produktov alebo trhov.

Product Management – používa sa vtedy, ak má individuálna osoba formálnu autoritu a právomoc rozhodovať o všetkých aspektoch plánovania, produkcie výrobkov, inventáru, distribúcie a predaja. Výrobný manažér koordinuje a urýchljuje požiadavky výroby, distribúcie a predaja ako nepretržitého toku výrobku až po jeho dodanie zákazníkovi.

Ad hoc Management – pre mnoho projektov, špeciálne menšieho rozsahu alebo kratšieho trvania, je síce potrebné zostaviť dočasný tím, obyčajne však vo funkčnom útvare alebo ako separátny organizačný „podúťvar“. Tieto tímy, ktoré nazývané ad hoc komisie, sú nezávislé na čele s jednou osobou [12].

2.2.3 Predpoklady úspechu projektového manažmentu

Podľa rôznych zdrojov ktoré sú uvedené v literatúre, aby bolo možné optimalizovať využitie projektového manažmentu, je potrebné preskúmať nasledujúce faktory [12]:

Potrebnosť projektu

Z hľadiska tohto faktora si treba položiť otázky:

- Je cieľ taký dôležitý, aby sa zradila dvojité podriadenosť pracovníkov, a tým zvýšené náklady?

- Môžu útvary, ktoré poskytnú pracovníkov projektovému útvaru, ďalej zabezpečiť požadované výkony?

Funkcia iniciátorov

Ten, kto zakladá projektovú skupinu, musí s ňou udržiavať stály kontakt a poskytovať potrebnú pomoc a podporu. Musí presadzovať schopnosti a uznávať autonómiu skupiny. Na druhej strane však musí zasahovať, ak sa pri voľbe (výbere) vyskytnú chyby v rozhodovaní.

Príprava kooperácie

Je potrebné sa rozhodnúť spolu so súčasnými vedúcimi útvarov, ktorých pracovníkov vybrať a koľko pracovných hodín sa majú venovať prácam na projekte. Keď sa to podarí, môže sa tým ušetriť rad kontroverzií, ktoré nežne vznikajú pri spracúvaní projektov.

Tvorba projektovej skupiny

V značnej miere závisí od stanovenej úlohy. Ktorá charakteristika – napr. veľkosť voľba kritérií – bude použitá, závisí od konkrétnej úlohy. Pritom musia byť zjednotené individuálne prednosti každého člena z hľadiska špecifickosti projektov. Predovšetkým sa musia zohľadňovať dve základné charakteristiky: veľkosť skupiny a kvalifikácia vybratých pracovníkov.

Roly

Značnou výhodou projektovej skupiny je, keď sa jej členovia dokážu stotožniť s určitými rolami, ktoré sa vyvíjajú v každej skupine. Sú to roly: vedúci, aktivizátor, tvrdohlavec, analytik, rojko, pedant, lojálny pracovník a poradca. Význam jednotlivých rolí a ich nositeľov sa uplatňuje v rôznych fázach projektového manažmentu.

Kontrola napredovania

Zabezpečuje ju iniciátor v úzkej koordinácii s vedúcim projektu, tak z kvantitatívnej, ako i kvalitatívnej stránky.

Ukončenie projektovej úlohy

Je potrebné presne a zrozumiteľne určiť termín ukončenia prác. Ak sa takéto rozhodnutie odkladá, chýba alebo sa porušuje, pôsobí to negatívne na pracovnú skupinu aj iniciátora.

Úspech uplatnenia projektového manažmentu však úzko súvisí aj s druhom projektu. Projekt je zvyčajne významná jednorazová akcia, ktorá sa vymyká bežných vstupov do podniku a vyžaduje zvláštne organizačné usporiadanie. Tým projekt sústreďuje na seba mimoriadnu pozornosť. Neúspech projektu má preto vysokú publicitu a závažné dôsledky [5].

S riešením každého projektu je spojené značné riziko. Príprava projektu potrebuje mimoriadne opatrenia, ktoré vyžadujú čas a náklady. Tieto sú v prípade neúspechu projektu stratené. Okrem finančných materiálnych škôd vznikajú aj nepríjemné dôsledky na prestíž zúčastnených vedúcich a výkonných pracovníkov. Za týchto podmienok môže byť strach z neúspechu bariérou angažovanosti i efektívnosti projektu [11].

3 Procesy riadenia projektov

V procese riadenia projektov sa snažíme vykonávať preventívne činnosti predchádzajúce problémom. Na základe projektového plánu a výsledkov práce, príp. iných záznamov o projekte zostavujeme správy o výkone, pomocou ktorých kontrolujeme vývin projektu. Môže sa jednať o riadenie času, kvality, nákladov, rizika alebo zmien projektu. Taktiež na začiatku každého riadenia je overenie realizovateľnosti projektov [8].

3.1 Overenie realizovateľnosti projektov

Na posudzovanie realizovateľnosti projektov sa využíva rada metód, pomocou ktorých je možné analyzovať a porovnávať jednotlivé fázy, alebo etapy projektov. Metódy posudzovania realizovateľnosti projektov môžu byť napríklad [5]:

- projektové príležitosti (Project Opportunity),
- predbežná (úvodná) štúdia realizovateľnosti (PFS - Pre Feasibility Study),
- predbežná (úvodná) štúdia financovania (Pre Investment Study),
- štúdia realizovateľnosti (FS - Feasibility Study),
- projekt financovania (Financing Project),
- analýza finančného toku (Cash Flow Analysis),
- analýza zisku (Ratio Analysis), a iné.

Medzi najpoužívanjšie metódy posúdenia realizovateľnosti projektov patrí predbežná štúdia možnosti realizovateľnosti.

Predbežná štúdia možnosti realizovateľnosti je úvodná štúdia, ktorá slúži na prvotné, variantné, časovo a finančne nenáročné posúdenie projektového zámeru pre jeho prijatie, úpravu alebo zamietnutie (vyradenie). Predbežná štúdia možnosti realizovateľnosti môže obsahovať viac variantov riešenia projektov. Predbežná štúdia realizovateľnosti odpovedá na otázky:

1. **prečo?** - Prečo sa projekt robí, aký je jeho zámer, ako naplňa víziu a poslanie?
2. **čo?** - Čo sa musí urobiť? Aké sú ciele?
3. **ako?** - Ako sa má postupovať, aké metódy a techniky použiť?
4. **kde?** - Kde sa projekt uskutoční?
5. **kto?** - Ktoré organizácie, oddelenia a jednotlivci sa zúčastnia prípravy, financovania a realizácie projektu?
6. **kedy?** - Kedy sa projekt začne a kedy má byť zakončený?
7. **koľko?** - Koľko bude projekt stáť?

Praktickým výsledkom predbežnej štúdie realizovateľnosti je výber projektového zámeru, resp. alternatívy, ktorá sa stane predmetom detailných analýz v rámci vypracovania štúdie realizovateľnosti.

Pri zložitých, komplexných alebo špeciálnych projektoch slúži predbežná štúdia realizovateľnosti na rozhodnutie, či ďalej pokračovať v príprave alebo návrh projektu zamietnuť, pretože následná štúdia realizovateľnosti je náročná a finančne nákladná.

Pri jednoduchých projektoch môže postačovať na schválenie projektu a pokračovanie vo fáze detailného plánovania [8].

3.2 Riadenie času

Vo fáze realizácie projektu, má projektový tím vykonávať pravidelné sledovanie stavu časového plánu projektu. Má sa analyzovať postup projektu, aby sa identifikovali trendy a možné neistoty vo zvyšnej práci na projekte.

Pri hodnoteniach priebehu sa majú používať platné časové plány. Majú sa identifikovať a analyzovať odchýlky od časového plánu, a ak sú významné, má sa na ne reagovať [2].

Majú sa identifikovať kľúčové príčiny odchýlok od časového plánu, či už priaznivé, alebo nepriaznivé. Príčiny priaznivých a nepriaznivých odchýlok sa majú využiť, aby poskytli údaje pre trvalé zlepšovania. Majú sa určiť možné vplyvy zmien časového plánu na rozpočet a zdroje pre projekt a na kvalitu projektu. Rozhodnutia o opatreniach, ktoré treba prijať, sa majú urobiť na základe faktov a po zvážení ich dôsledku pre ďalšie procesy a ciele projektu.

Riadenie času zahŕňa tieto kroky [4]:

- spätná väzba od zainteresovaných (výsledky monitorovania), plnenie časového plánu s dôrazom na kritickú reťaz,
- vyhodnotenie odchýlok od plánovaných termínov, identifikácia príčin omeškania, návrh a zavedenie nápravných opatrení,
- aktualizácia časového plánu (úsečkový graf, sieťový graf),
- zdokumentovanie zmeneného stavu.

Príčinami odchýlok času môžu byť najmä snahy o technickú dokonalosť ktoré ohrozujú termín, alebo keď sa časový plán musí zmeniť zmenia aj požiadavky na špecifikáciu realizácie (práca navyše spôsobuje sklz).

Predísť odchýlkam času je možno hlavne prevenciou a s tým, že zabránime zmenám v projekte.

3.3 Riadenie kvality

Kvalita je súhrn charakteristík objektov, ktoré sa týkajú schopnosti vyhovieť daným a predpokladaným potrebám.

Riadenie kvality obsahuje všetky činnosti v rámci všetkých funkcií riadenia, ktoré definuje stratégia kvality, ciele a zodpovednosť. Tieto činnosti sú uvádzané do praxe pomocou plánovania kvality, kontroly kvality, zabezpečenia kvality a zlepšenia kvality v rámci systému riadenia kvality.

Systém riadenia kvality zahŕňa najmä postupy, metódy a zdroje, potrebné pre zavádzanie systému riadenia kvality.

Riadenie kvality projektov zahŕňa tieto kroky [8]:

- spätná väzba od zainteresovaných (výsledky monitorovania), plnenie SMART cieľov a kritických faktorov spokojnosti všetkých záujmových skupín,
- aktualizácia kvalitatívnych ukazovateľov, SMART cieľov, kritických faktorov spokojnosti záujmových skupín,
- zdokumentovanie zmeneného stavu.

Príčinami odchýlok kvality sú najmä:

- nedostatočná komunikácia,
- predpoklady jednej strany sú príliš optimistické,
- chyba dodávateľa v realizačnej fáze dodávky.

Predísť odchýlkam kvality je možné prevenciou, pomocou kvalitne a podrobne spracovanej zmluvy a špecifikácie. Dôležitou vlastnosťou je pri budovaní kvality priebežná komunikácia zo zákazníkom [11].

3.4 Riadenie nákladov

Pred akýmkoľvek výdavkami sa má vytvoriť a zdokumentovať systém riadenia nákladov a súvisiace postupy a má sa predložiť tomu, kto zodpovedá za prácu alebo výdavky [8].

Pre potreby riadenia nákladov sa má určiť časový plán sledovanie stavu, frekvencia zberu údajov a predpovedí. To zabezpečuje primerané riadenie činnosti v rámci projektu a príslušných informácií. Projektový tím má overiť, či sa v rámci zostávajúceho rozpočtu dajú zvyšné práce vykonať a skompletizovať.

Majú sa analyzovať trendy nákladov na projekt pomocou takých techník, ako je napr. analýza pridanej hodnoty. Má sa preskúmať plán zvyšných prác, aby sa identifikovali neistoty. Majú sa identifikovať kľúčové príčiny odchýlok od rozpočtu, či už priaznivé alebo nepriaznivé. Má sa prijať opatrenie na ubezpečenie, že nepriaznivé odchýlky neovplyvnia ciele projektu.

Projektový tím má pravidelne vykonávať preskúmanie nákladov na projekt, ako sa to uvádza v pláne manažmentu projektu a má brať do úvahy ďalšie finančné preskúmania. Príčinami odchýlok nákladov projektu sú najmä:

- znižovanie nákladov bez zmeny časového plánu alebo kvality,
- zákazník stláča náklady zneužitím konkurencie (súťaž klamárov),
- neznalosť (chyby) pri kalkulácii nákladov, zlý systém prevádzkového účtovníctva.

Predísť odchýlkam nákladov môžeme prevenciou, kvalitne a podrobne spracovanou finančnou analýzou a rozpočtom, alebo s tým, že zabránime zmenám v projekte [7].

3.5 Riadenie zmeny

Zmeny, ktoré majú účinok na ciele projektu sa majú odsúhlasiť so zadávateľom projektu a s ďalšími príslušnými zainteresovanými stranami. Riadenie zmien má zohľadňovať aj [8]:

- riadiace zmeny predmetu projektu, cieľov projektu a plánu riadenia projektu,
- koordinačné zmeny medzi previazanými procesmi projektu a pri riešení akýchkoľvek konfliktov,
- postupy dokumentovania zmeny,
- aspekty zmeny ovplyvňujúce pracovníkov.

Zmeny môžu mať pozitívne účinky (príležitosti), ale aj záporné účinky na projekt a preto sa majú identifikovať čím skôr. Majú sa analyzovať kľúčové príčiny záporných účinkov a výsledky sa majú využiť na vypracovanie preventívnych opatrení a zavedenie zlepšenia. Nech sa zmena týka čohokoľvek, je potrebné na ňu správne zareagovať. Pri požiadavke na zmenu treba identifikovať, čoho sa bude zmena týkať a v akom rozsahu zasiahne už do vykonanej práce. Čím neskôr v priebehu projektu sa má zmena vykonať, tým bude jej ošetrovanie náročnejšie. Na základe projektového plánu, správ o výkone a požiadavky na zmenu sa vykoná zapracovanie zmeny do plánu projektu a určí sa opravná akcia [4].

4 Riadenie rizík v IT projektoch

Všetky procesy riadenia projektov sú dôležité a nemali by sme vynechávať ani jeden. Riadenie zmien obísť nemôžeme, pretože keď zákazník vyžaduje zmenu alebo došlo k požiadavke na zmenu z vnútra organizácie, musíme to nejako ošetriť, aby sme mohli ísť ďalej. Riadenie rizík je však činnosť, ktorá je na prvý pohľad v procese riadenia niečím extra, ktorá síce môže pozitívne ovplyvniť celkový výsledok projektu, ale nie je nutná pre úspešné dokončenie projektu.

Opak je však pravdou. V praxi sa často stáva, že manažér projektu pokladá riadenie rizík za zbytočné zdržovanie ho od „dôležitejšej“ roboty. A tak práve manažment rizík je v procese riadenia pre mnohé projekty osudným.

Identifikovanie rizík na začiatku projektu je dôležité, avšak riadenie rizík počas celého projektu má na úspech projektu silný vplyv. Z výskumu vyplýva, že podľa toho, či sú riziká manažované počas celého projektu, sme schopní predpokladať výsledok projektu až na 69%. Z neúspešných projektov riziká počas celého projektu manažovalo iba 20% a z úspešných až 60%. Manažment rizík je ukazovateľom kvalitného projektového manažmentu [6].

4.1 Definícia pojmu riziko

Riziko je historický výraz, pochádzajúci údajne zo 17. storočia, keď sa objavil v súvislosti s loďnou plavbou. Výraz „risico“ pochádza z taliančiny a označoval úskalie, ktorému sa museli plavci vyhnúť. Podľa dnešných výkladov sa rizikám obecné rozumie nebezpečie vzniku škody, poškodenia, stratenia alebo zničenia, prípadne nezdaru pri podnikania [10].

Pojem **riziko** teda nadväzuje na filozofické kategórie, akými sú nutnosť a náhodnosť. V ekonómii je pojem „riziko“ používané v súvislosti s nejednoznačnosťou priebehu určitých skutočných ekonomických procesov a nejednoznačnosťou ich výsledku. Obecné je možné samozrejme konštatovať, že nemusí ísť len o riziko ekonomické.

Riziko je teda často chápané ako nebezpečie vzniku určitej straty. Finančná teória obvykle definuje riziko ako volatilitu (kolísavosť) finančnej veličiny okolo očakávanej hodnoty v dôsledku zmien rady parametrov. Riziko je treba hodnotiť z dvoch strán [10]:

- **pozitívne stránky** – nádeje vyššieho zisku, nádeje vyššieho úspechu,
- **negatívne stránky** – nebezpečie horších výsledkov.

Straty môžu vzniknúť prostredníctvom kombinácie týchto faktorov – volatility finančných premenných ovplyvňujúcich mieru rizika a celkovej angažovanosti k týmto zdrojom.

4.2 Význam riadenia rizík v projektoch

Riadenie rizík v projektoch je z časti exaktnou vedou a z časti umením, ktoré zahŕňa identifikáciu rizík, ich analýzu a reakciu na ňu po celú dobu života projektu. Pritom nesmieme zabudnúť splniť si najlepšie záujmy pre naplnenie cieľov projektu. Riadenie rizík je jedným z často zabudnutých aspektov pri riadení projektov, ale pri správnom uplatnení môže naopak veľmi výrazne prispieť k zvýšeniu konečnej úspešnosti projektov. Správne riadenie rizík môže mať pozitívne dopady na výber projektov, stanovenia rozsahu, aj do vytvorenia realistického časového plánu a odhadov nákladov.

Dobré riadenie rizík v projektoch prebieha často celé bez povšimnutia, tak isto ako krízové riadenie. Pri krízovom riadení vzniká priame nebezpečenstvo pre úspech projektov a preto kríza na seba upúta veľmi výraznú pozornosť celého projektového tímu. Vyriešenie krízy je tak isto podstatne lepšie viditeľné ako úspešné riadenie rizík. Efektívne riadenie rizík ale znamená zníženie počtu problémov a tým pádom urýchlenie ich riešení (menej problémov zvládneme vyriešiť rýchlejšie). Vonkajší pozorovateľ často ale nepozná, že hladkým vývojom nového systému stálo dobre odvedené riadenie rizík alebo „len“ súhrn šťastných náhod, ale samotný projektový tím vždy dobre vie, že sa konkrétny projekt podarilo vďaka dobrému riadeniu rizík.

Výskumná spoločnosť KLCI Research Group previedla v roku 2001 prieskum medzi 260 organizáciami zamarených na vývoj softwaru po celom svete a sledovala praktické riadenie rizík vo vývoji softwaru. Tu je niekoľko výsledkov [1]:

- 97 percent účastníkov potvrdilo, že majú zavedené postupy pre identifikáciu a ohodnotenie rizík,
- 80 percent účastníkov uviedlo, že hlavným prínosom riadenia rizík je predpovedanie a zabránenie problémom,
- 70 percent organizácií malo definované procesy vývoja softwaru,
- 64 percent organizácií malo ustanovenú kanceláriu alebo útvar riadenia projektov.

Celá rada spoločností dokázala prežiť na trhu len vďaka tomu, že na seba vzala určité riziko, za ktorým ale stály obrovské príležitosti. Niektorí odborníci na riziká doporučujú, aby sa organizácia aj jednotlivci vo všetkých stránkach projektov aj vo svojich osobných životoch pokúsili nájsť vhodnú rovnováhu medzi rizikami a príležitosťami. Myšlienka tohto dosadenia rovnováhy medzi rizikami a príležitosťami nám povie, že rôzne organizácie a rôzni ľudia budú mať k rizikám rôznu toleranciu. Sú organizácie a ľudia, ktorí majú k rizikám neutrálnu toleranciu, iný k nim majú odpor, a iný riziká s obľubou vyhľadávajú. Tieto tri typy preferencií rizika sú súčasťou teórie užitku rizík [12].

Úžitok z rizika alebo tolerancia k rizikám je množstvo uspokojenia alebo výhody, plynúce z potenciálnych prínosov rizika.

Na **cieľ riadenia rizík** v projektoch môžeme pozeráť ako na minimalizáciu potenciálnych negatívnych rizík za súčasné maximalizácie potenciálnych pozitívnych rizík. Riziká, ktoré už projektový tím identifikoval a analyzoval, sa niekedy označuje pojmom „známe riziko“. Tieto známe riziká môžeme riadiť aktívne. Neznáme riziká, ktoré sme neidentifikovali ani neanalyzovali, ale riadiť nemôžeme. Preto je známe, že dobrý projektový manažér vie, ako dôležité je najprv identifikovať riziko v projekte a potom riadiť. Súčasťou riadenia rizík je celkom šesť hlavných procesov. Tieto procesy popíšem v nasledujúcej kapitole diplomovej práce [1].

4.3 Plánovanie riadenia rizík

V procese plánovania riadeniu rizík sa musíme rozhodnúť, akým spôsobom pristupovať k aktivitám riadenia rizík v projekte a ako ich plánovať. Pri tomto plánovaní prevedie projektový tím revíziu stanovenia rozsahu projektu, plánu riadenia projektu, faktorov podnikového prostredia a organizačných aktív organizácie, a tým pádom môže rozobrať a analyzovať aktivity riadenia rizík v konkrétnom projekte. Hlavným výstupom z tohto procesu je takzvaný plán riadenia rizík, ktorý dokumentuje postupy riadenia rizík v projekte

Plán riadenia rizík

Projektový tím by mal plán riadenia rizík vytvoriť najlepšie na sérii porád, usporiadaných hneď na začiatku životného cyklu projektu. Pri týchto prácach musí projektový tím zrevidovať projektové dokumenty a podnikové zásady či politiky riadenia rizík, kategórie rizík, správy s poučením z predchádzajúcich projektov. Dôležité je overiť si, akú toleranciu k rizikám vykazujú jednotliví účastníci projektu.

Plán riadenia rizík zahŕňa spôsob riadenia rizík v konkrétnom riešenom projekte. Podobne ako plány z ostatných oblastí poznatkov pre riadenie projektov je aj tento plán súčasťou celkového plánu riadenia. V tomto pláne je dôležité vyjasniť si roly jednotlivých členov a ich povinnosti, pripraviť si odhad rozpočtu a časového plánu na práce, ktoré súvisia s rizikami, a identifikovať zvažované kategórie rizík. Ďalej tu musíme popísať, akým spôsobom budeme robiť vlastné riadenie rizík, teda ako budeme mimo inej hodnotiť pravdepodobnosti a dopady rizík a ako vytvárať dokumenty s rizikami. Úroveň detailu, s akou budeme plán riadenia rizík vytvárať, sa môže líšiť podľa potrieb konkrétneho projektu. Okrem plánu riadenia rizík býva vo veľa projektoch nutné vytvoriť si aj mimoriadne plány [2].

Mimoriadny plán

Mimoriadne plány sú predtým definované postupy, ktorými sa bude projektový tím riadiť v prípade vzniku rizikovej udalosti. Pretože napríklad projektový tím vie, že určitá verzia softwarového balíku, s ktorým chcú v projekte pracovať, nemusí v potrebnom okamžiku byť k dispozícii, môžu si vytvoriť vhodný mimoriadny plán, v rámci čoho rozhodnú o používaní staršej verzii takého istého softwaru.

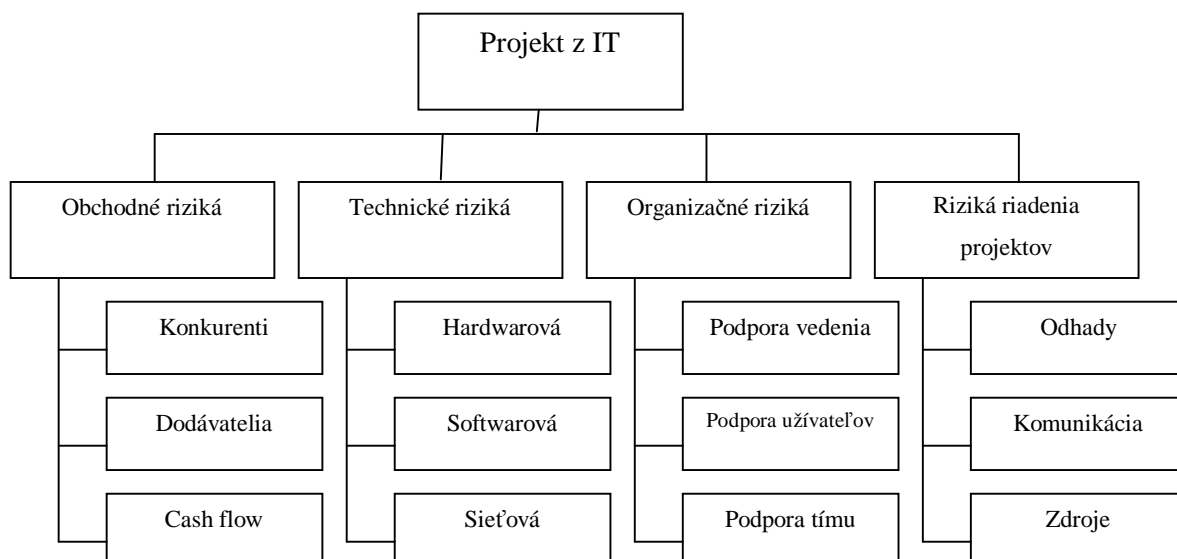
Havarijný plán

Havarijné plány sú pripravené pre zvládnutie rizík s vysokými dopadmi na splnenie cieľov projektu. Tieto plány sa realizujú v prípade, že pokusy o potlačenie rizika nie sú účinné.

Niekedy sa dokonca pojmy mimoriadnych, havarijných, prípadne núdzových plánov voľne zmeňujú. **Mimoriadne rezervy** alebo **mimoriadne prírážky** sú potom opatrenia, pomocou ktorých zadávateľ projektu či organizácie znižuje na prijateľnú mieru riziko prekročenia nákladov alebo časového plánu projektu.

Dobrou metódou pre pochopenie najbežnejších zdrojov rizík v projektoch z informačných technológií je napríklad revízia navrhovaného projektu podľa kritérií úspechu, dotazníku rizík, prípadne podľa iného podobného nástroja. Užitočné je aj pozrieť si na štruktúru rozpisu práci v projekte a skontrolovať, že niektoré z kategórií štruktúry práce zo sebou neprinášajú určité konkrétne riziká. Keď je napríklad úlohou jednej položky v štruktúre práce vytvorenie tlačovej správy, pritom tlačovú správu nikto v projektovom tíme doposiaľ nezostavoval, môže sa vzniknúť negatívne riziko jej neprofesionálneho spracovanie.

Užitočným nástrojom, pomocou ktorého môže projektový manažér posúdiť potenciálne riziká v rôznych kategóriách, je **štruktúra rozdelenia rizík** – čiastočne sa podobá štruktúre rozdelenia či rozpisu práce a definuje hierarchiu kategórií potenciálnych rizík v projekte. Príklad štruktúry rozdelenia rizík, ktoré by sme mohli uplatniť v rade projektov z informačných technológií, je na obrázku 4.1 [1].



Obrázok 4.1:Príklad rozdelenia štruktúry rizík [1]

Na najvyššej úrovni sú tu vyznačené kategórie rizík obchodných, technických, organizačných a z projektového riadenia. Pod obchodné riziká spadajú podkategórie rizík konkurentov, dodávateľov a nepriaznivých finančných tokov. Ako technické riziká sú zvažované hardwarové, softwarové a sieťové riziká. Štruktúra rozdelenia rizík prehľadne, graficky a na jednej strane šikovne zhrňuje všetky dôležité kategórie rizík, ktoré súvisia so všetkými projektmi z informačných technológií.

Okrem identifikácie rizík, povené na základe povahy konkrétneho projektu alebo produktu, ktoré majú byť jeho výsledkom, je tak isto dôležité identifikovať potenciálne riziká v súlade s jednotlivými oblasťami poznatkov z riadenia projektov, ako je rozsah čas, náklady a kvalita.

Porozumieť najbežnejším zdrojom rizík je veľmi užitočné pri procese identifikácie rizík, ktorý je ďalším krokom riadenia rizík v projektoch [4].

4.4 Identifikácie rizík

V rámci procesu identifikácie rizík je potrebné pochopiť, aké potenciálne udalosti môžu poškodiť alebo vylepšiť stav riešenia konkrétneho projektu. Potenciálne riziká je dôležité identifikovať (rozpoznať) včas, ale na druhej strane musíme riziká identifikovať aj priebežne, podľa potreby meniaceho prostredia projektu. Keď sa riziko nepodarí identifikovať nie je možné ho ani správne riadiť.

4.4.1 Námety pre identifikáciu rizík

Pre identifikáciu rizík existuje niekoľko rôznych nástrojov a techník. Projektové tímy často začínajú tento proces revíziou projektových dokumentácií, aktuálnych a historických informácií, ktoré môžu mať vplyv na riešenie projektu. Členovia projektových tímov diskutujú o týchto informáciách na poradách, a kladú tu dôležité otázky ohľadne vzťahu preberaných faktorov k rizikám. Keď projektový tím na úvodnej porade identifikuje základný okruh potenciálnych rizík, môže pomocou rôznych techník zhromažďovania informácií identifikovať ďalšie riziká. Najdôležitejšie techniky zhromažďovania informácií sú [1]:

Brainstorming

Brainstorming je technika, pri ktorej sa určitá skupina pokúsi „vyrobiť“ nápady alebo nájsť postup pre riešenie určitého problému takým spôsobom, že jednotliví členovia prinášajú nápady najprv spontánne, bez okamžitého kritického posudzovania. Diskusiu formou brainstormingu by mal viesť skúsený moderátor, ktorý by mal prednášať aj nové kategórie potenciálnych rizík a priebežne tak oživovať tok myšlienok. Po zhromaždení nápadov môže moderátor výsledky zoskupiť a rozdeliť do lepšie zvládnuteľných kategórií.

S metódou brainstormingu je ale nutné zachádzať opatrne a rozhodne by sme ju nemali používať príliš, alebo nesprávne. Vymyslenie nových nápadov niekedy u účastníkov oslabujú skupinové efekty, ako je napríklad strach zo spoločenského znemožnenia.

Delfská metóda

Základná myšlienka delfskej metódy je dosiahnutie zhody v skupine odborníkov (expertov), ktorí sa pokúšajú predvídať budúci vývoj. Delfská metóda sa prevadí v opakovaných kolách otázok a písomných odpovediach, medzi ktoré môžu patriť i pripomienky k odpovediam z predchádzajúcich kôl. Takto je možné zhromaždiť si podklady od celej skupiny a zároveň sa vyhnúť jednostrannému zaujatiu. Pri uplatnení delfskej metódy musíme ale zvoliť vhodnú skupinu odborníkov na dotknutú oblasť.

Rozhovor

Rozhovor je technika hľadania faktov a zhromažďovania informácií, ktorá správne prebieha tvárou tvár, ale často k nej môže patriť aj telefónny hovor, elektronická pošta alebo rýchla výmena správ. Dotazovanie a vedenie rozhovoru s ľuďmi, ktorí majú skúsenosti s podobnými projektmi, sú mimo iného dôležitým nástrojom pre identifikáciu rizík. Keď máme napríklad v rámci riešenia projektu pracovať s nejakým konkrétnym typom hardwaru alebo softwaru, môžeme osloviť niekoho, kto má čerstvé skúsenosti s týmto hardwarom a softwarom.

Analýza SWOT

Analýza SWOT je samotná analýza silných stránok, slabých stránok, príležitostí a hrozieb, ktorá sa často používa pri strategickom plánovaní. Analýzu SWOT môžeme ale používať aj v rámci identifikácie rizík, kde sa projektový tím sústreďí na širšie súvislosti potenciálnych rizík v konkrétnych projektoch. Keď analýzu SWOT aplikujeme na konkrétne potenciálne projekty, môžeme s tým identifikovať širšie riziká a príležitosti daného scenára.

Schéma SWOT analýza je zobrazená na obrázku 4.2.

	S - silné stránky	W - slabé stránky
O - príležitosti	Strategie SO	Strategie WO
T - hrozby	Strategie ST	Strategie WT

Obrázok 4.2: Schéma SWOT analýzy [3]

Na obrázku 4.2 je znázornený tabuľka kde stratégia **SO** znamená využitie silnej stránky na získanie výhod, stratégia **WO** sa uplatňuje na prekonanie slabostí využitím príležitosti, pri stratégii **SW** sa využívajú silné stránky na čelenie hrozbám a stratégia **WT** znamená minimalizovanie nákladov a čelenie hrozbám.

Ďalšími troma technikami pri identifikácii rizík sú kontrolné zoznamy, analýza predpokladov a vytváranie rôznych diagramov alebo grafov [2].

4.4.2 Register rizík

Hlavným výstupom z procesu identifikácie rizík je zoznam identifikovaných rizík a ďalšie informácie, ktoré nesmú chýbať pre začiatok tvorby takzvaného **registru rizík**. Je to dokument, ktorý obsahuje výsledky rôznych procesov riadenia rizík a často býva znázornený vo forme tabuľky alebo tabuľkového listu. Je to nástroj pre dokumentovanie potenciálnych rizikových udalostí a s nimi súvisiacich informácií.

Rizikové udalosti sú pritom konkrétne neisté okolnosti alebo udalosti, ktoré môžu vzniknúť a poškodiť projekt alebo mu prospieť. Medzi negatívne rizikové udalosti patrí napríklad nedosiahnutie výkonných parametrov určitého výrobku, vytvoreného v rámci projektu, meškanie pri dokončení práce oproti časovému plánu, zvýšenie odhadovaných nákladov, nedostatok materiálu pre výrobu, stávky a ďalšie. Ako pozitívne riziká môžeme brať dokončenie práce za kratšie ako naplánovaná doba alebo za menšie náklady ako je naplánované, spolupráca s dodávateľmi pre návrh lepších výrobkov.

Register rizík by mohol obsahovať nasledujúce informácie [1]:

- identifikačné číslo každej rizikovej udalosti,
- hodnotenie danej rizikovej udalosti,
- názov rizikovej udalosti,
- popis rizikovej udalosti,
- kategória, do ktorého táto riziková udalosť spadá,
- prvotnú príčinu rizika,
- spúšťače danej rizikovej udalosti,
- potenciálne reakcie na jednotlivé riziká,
- vlastníka rizika,
- pravdepodobnosť vzniku rizika,
- dopady prípadného vzniku rizika pri riešení projektu,
- stav rizika.

Po identifikácii rizík môžeme pristúpiť k ďalšiemu kroku, pri ktorom prevedieme kvalitatívnu analýzu rizík a pokúsime sa pochopiť, aké riziká sú najväčšie.

4.5 Kvalitatívna analýza rizík

Súčasťou kvalitatívnej analýzy rizík je hodnotenie pravdepodobnosti a dopadu jednotlivých identifikovaných rizík, pri ktorom stanovíme ich rozmer a prioritu. V ďalších podkapitolách budú popísané tieto metódy a ich dopady na analýzu rizík [1].

4.5.1 Výpočet rizikových faktorov pomocou matice pravdepodobnosti a dôsledkov

Ľudia často o pravdepodobnosti alebo dôsledkov rizika jednoducho povedia, že sú vysoké, stredné, priemerné alebo nízke.

Projektový manažér môže pravdepodobnosť rizika a ich dôsledkov znázorniť v takzvanej **matici alebo diagramov pravdepodobností a dôsledkov** (obrázok 4.3)– je to matica alebo diagram, ktorá znázorňuje na jednej strane matici alebo na jednej osi grafu relatívnu pravdepodobnosť vzniku rizika a na druhej relatívne dôsledky vzniku rizika. Pri tomto postupe musia účastníci projektu najprv uviesť riziká, ktoré podľa ich názorov v projekte hrozia. Ďalej každé riziko sa označí ako vysoké, stredné alebo nízke podľa pravdepodobnosti vzniku a zhodnotí sa, či môže riziko vyvolať vysoké, stredné alebo nízke dôsledky [1].

Zistené výsledky zhrnie manažér projektu do matice alebo diagramu pravdepodobnosti a dôsledkov. Kvalitatívna analýza rizík prebieha normálne dosť rýchlo, takže projektový tím sa musí včas rozhodnúť, aký postup má v danom projekte najväčší zmysel. Niekedy môže byť užitočné vytvoriť si samostatnú maticu alebo diagram pravdepodobnosti a dôsledkov aj pre negatívne a pre pozitívne riziká aby sme na žiadnu nezabudli.

Pravdepodobnosť	Vysoká	Riziko 6	Riziko 9	
	Stredná	Riziko3 Riziko7	Riziko 5 Riziko 2	Riziko 1 Riziko 4
	Nízka			Riziko 8
		Nízke	Stredné	Vysoké
		Dôsledky		

Obrázok 4.3:Príklad matice pravdepodobnosti a dôsledkov [1]

Niektoré projektové tímy vypočítajú ku každému riziku jediné číselné skóre, ktoré vznikne vynásobením číselného skóre pravdepodobnosti a číselného skóre dôsledku. Dôsledný postup vypočítava z informácie o pravdepodobnosti a dôsledkov takzvané **rizikové faktory** – to sú veličiny, ktoré vyvinula americká Vysoká škola riadenia systému obrany a vyjadrujú celkové riziko konkrétnych udalostí pri danej pravdepodobnosti ich vzniku a pri daných dôsledkov pre projekt. Uvedená technika teda využíva údaje z matíc pravdepodobnosti a dôsledkov.

Pravdepodobnosť vzniku rizika môžeme odhadnúť na základe faktorov, ktoré odvodím od jedinečnej povahy konkrétného projektu. Medzi faktory posudzované u potenciálnych technologických rizík z kategórie hardwaru a softwaru môže napríklad patriť nedostatočná vyspelosť použitej technológie, príliš veľká zložitosť technológie, alebo nezodpovedajúca podpora vývoja technológie. Ako dopady skutočného vzniku rizika môžeme zvažovať napríklad dostupnosť prípadných havarijných riešení alebo hroziace dôsledky pri nesplnení pôvodných odhadov efektivity, nákladu a časového plánu [10].

4.5.2 Sledovanie desiatich najrizikovejších položiek

Táto metóda je nástrojom kvalitatívnej analýzy rizík, ktorej úlohou je okrem identifikácie rizík aj udržiavanie podvedomia o rizikách po celú dobu života projektu. Jej súčasťou je zavedenie pravidelných revízií zoznamu najvýznamnejších rizikových položiek v projekte zo spoluprácou s vedením prípadne aj so zákazníkom. Revízia sa začína zhrnutím stavov desiatich najzávažnejších zdrojov rizika v projekte, v ktorom je u každej položky uvedené aktuálne hodnotenie rizika, predchádzajúce hodnotenie, koľkokrát sa objavilo na zozname v priebehu času a aký postup sa dosiahol pri riešení tejto rizikovej položky od poslednej revízie.

Revízie riadenia rizík naplňuje súčasne hneď niekoľko cieľov. Za prvé, vedenie firmy a zákazník sú priebežne informovaní o najdôležitejších vplyvoch, ktoré by mohli podporiť alebo naopak ohroziť úspech projektu. Za druhé, pri zapojení zákazníka môže projektový tím zvažovať aj iné alternatívne stratégie riešenia rizík. A za tretie sa jedná o prostriedok pre posilnenie dôvery v projektovom tíme, pretože firemnému vedeniu aj zákazníkovi jasne ukážeme, že si projektový tím je vedomí všetkých významných rizík a má prijatú vhodnú stratégiu [1].

4.5.3 Expertné posudzovanie

Celá rada organizácií sa pri prevedení kvalitatívnej analýzy rizík spolieha najprv na intuitívne vedené názory a skúsenosti rôznych expertov či odborníkov. Toto expertné posudzovanie môžeme používať namiesto iných techník analýzy rizík, alebo spoločne s nimi ako doplnok. Experti môžu napríklad rozdeliť riziká do kategórie vysoké, stredné alebo nízke pravdepodobnosti a dôsledkov pri zapojení zložitejších techník, alebo bez týchto techník, čisto podľa svojho vlastného názoru.

Hlavným výstupom z kvalitatívnej analýzy rizík je aktualizovaná verzia registra rizík. To znamená, že v registri vyplníme (respektíve aktualizujeme) stĺpce s hodnotením rizík – zapíšeme do nej buď číselnú hodnotu, alebo hodnotenie vysoké/stredné/nízke pravdepodobnosti a dôsledky rizikovej udalosti. K rizikovým udalostiam je často nutné doplniť ešte rôzne ďalšie informácie, napríklad takto definujeme, ktoré v najbližšej dobe vyžadujú bližšiu pozornosť, alebo treba ich zaradiť na **sledovaný zoznam** (watch list).

Sledovaný zoznam je zoznam rizík, ktorý má nízku prioritu, ale aj tak je definované ako potenciálne riziko. Kvalitatívna analýza rizík môže aj identifikovať riziká, ktoré je treba podrobiť podrobnejšej kvantitatívnej analýze. Túto analýzu popisuje ďalšia kapitola tejto práce [1].

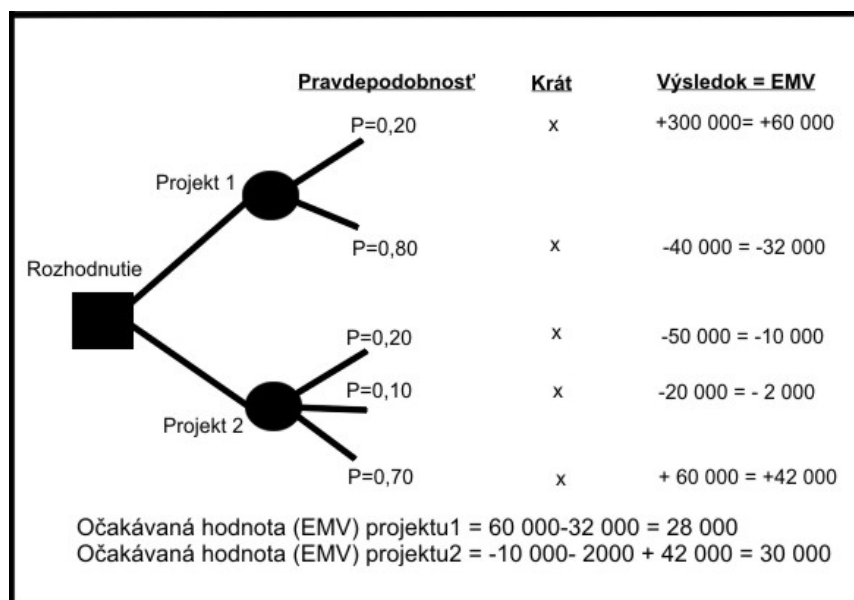
4.6 Kvantitatívna analýza rizík

Po kvalitatívnej analýze rizík nasleduje často kvantitatívna analýza, aj keď oba procesy môžu prebiehať aj spoločne, alebo naopak oddelene. V niektorých projektoch môže tímu stačiť len kvalitatívna analýza. Konkrétny typ prevedenia analýzy rizík vychádza z povahy daného projektu, ale aj z toho, či je na danú analýzu dost času a peňazí. U veľkých, zložitých projektov, do čoho sú zapojené nové špičkové technológie, býva často potreba previesť aj kvantitatívnu analýzu rizík.

Hlavnými technikami kvantitatívnej analýzy sú zhromažďovanie (zber) dát, vlastná kvantitatívna analýza a modelovanie. Súčasťou zberu dát bývajú rozhovory, expertné posudzovanie a zhromažďovanie informácií a rozdelenie pravdepodobností. Ďalšie podkapitoly popíšu niekoľko metód kvantitatívnej analýzy a modelovania rizík a analýzu rozhodovacieho stromu, simulácie a na analýzu citlivosti [1].

4.6.1 Rozhodovacie stromy a očakávaná peňažná hodnota

Rozhodovací strom je technika grafovej analýzy, ktorá napomáha pri výbere najlepšieho postupu v situáciách s neistými výsledkami. Jedným z bežných uplatnení analýzy rozhodovacieho stromu je výpočet **očakávanej peňažnej hodnoty** (EMV – expected monetary value) – čo zistíme ako súčin pravdepodobnosti rizikovej udalosti a peňažnej hodnoty tejto rizikovej udalosti. Na obrázku 4.4 je uvedený príklad rozhodovacieho stromu, podľa čoho sa organizácia rozhoduje o konkrétnych riešeniach [1].



Obrázok 4.4: Príklad výpočtu očakávanej peňažnej hodnoty [1]

Pri vytvorení rozhodovacieho stromu a pri výpočtu vlastnej očakávanej peňažnej hodnoty musíme najprv odhadnúť pravdepodobnosť, teda „šancu“ vzniku skúmaných udalostí. Na obrázku 4.3 napríklad vidíme, že úspešné uzavretie zmluvy na riešenie projektu 1 má pravdepodobnosť 20 percent ($P=0,20$), a keď zmluva bude skutočne uzavretá prinesie firme 300 000 dolárov zisku, čo je vyznačené ako výsledok projektu. Na druhej strane ale s pravdepodobnosťou 80 percenta ($P=0,80$) hrozí, že zmluva na riešenie projektu 1 uzavretá nebude, a v tejto situácii dosiahne firma výsledok 40 000 dolárov do prípravy ponuky [1].

4.6.2 Simulácia

Dômyselnejšou technikou kvantitatívnej analýzy rizík je simulácia. Táto metóda znamená, že na istej reprezentácie alebo modelu skúmame chovanie pôvodného „živého“ systému ktorý bol vzorom pre vytvorenie modelu.

Väčšina simulácií býva založená na určitej modifikácii metódy Monte Carlo. Metóda alebo analýza **Monte Carlo** je technika pre kvantifikáciu rizík, ktorá opakovane simuluje výsledky modelu a zisťuje tak štatistické rozdelenie vypočítaných výsledkov. Pomocou metódy Monte Carlo môžeme tak napríklad zistiť, že určitého dňa môže projekt skončiť len s pravdepodobnosťou 10 percent, zatiaľ čo iný deň má ako termín dokončenia projektu pravdepodobnosť 50 percent. Inými slovami, analýza Monte Carlo dokáže (mimo inej) predvídať pravdepodobnosť dokončenia projektu v určitý deň, alebo pravdepodobnosť, že náklady na projekt budú nižšie alebo rovné istej vopred stanovenej hodnoty. Pri prevedení analýzy Monte Carlo môžeme využívať radu rôznych distribučných funkcií [1].

4.6.3 Analýza citlivosti

Analýza citlivosti je technika ktorá zaisťuje akým spôsobom (do akej miery) sa zmeny jednej alebo viacej veličiny premietajú do konečných výsledkov. Pomocou analýzy citlivosti môžeme zaisťiť, aké mesačné splátky nás čakajú pri využití pôžičky na určitú čiastku pri rôznej dobe splatení alebo pri rôznej úrokovej miere.

Analýzu citlivosti využívajú tak isto mnohí profesionáli, ktorí s ňou nad rôznymi predpokladmi robia bežné typy podnikateľských rozhodnutí, napríklad analýza bodu zvratu.

Hlavnými výstupmi z kvantitatívnej analýzy rizík sú aktualizované verzie registru rizík. Kvantitatívna analýza podáva aj rámcové informácie o projekte, teda pravdepodobnosť dosiahnutia určitých rámcových cieľov projektu. Na základe týchto informácií potom môže manažér projektu navrhnúť zmeny v mimoriadnych rezervách [1].

4.7 Plánované reakcie na riziká

Akonáhle organizácia zaisťí, teda identifikuje riziká a podrobí ju kvantitatívnej analýze, musí na tieto riziká vytvoriť si vhodnú reakciu. Pri tvorbe reakcií na riziká definujeme možnosti a stratégie oslabenia negatívnych rizík a naopak posilnenie pozitívnych rizík. Na negatívne riziká môžeme zareagovať pomocou nasledujúcich stratégií [1]:

1. **Zabránenie rizika** - znamená potláčanie konkrétnej hrozby alebo rizika, dosiahnuté obvykle potláčaním jeho príčin. Samozrejme nie všetkým rizikám môžeme úplne zabrániť, ale konkrétnym udalostiam niekedy zabrániť môžeme.
2. **Prijatie rizika** – nie je nič iného ako prijatie (akceptovanie) dôsledku možného vzniku rizikovej udalosti.
3. **Prenos rizika** – je prenos dôsledkov rizika a zodpovednosti za jeho riadenie na tretiu stranu (subjekt). Tento typ reakcie na riziká sa veľmi často používa pri práci s finančnými rizikami.
4. **Potláčanie rizika** – vyjadruje zníženie dopadu rizikovej udalosti pomocou zníženia pravdepodobnosti jeho vzniku. Príkladom potlačenia rizika môže byť používanie osvedčených technológií, zapojenie odborne zdatného a preškoleného personálu a uplatnenie rôznych analytických a validačných techník.

Na pozitívne riziká môžeme zareagovať pomocou nasledujúcich 4 základných stratégií:

1. **Využitie rizika** – urobiť čokoľvek, aby príslušné kladné (pozitívne) riziko nastalo. Manažér projektu môže napríklad napísať článok do miestnych novín, sám napísať správu.
2. **Zdieľanie rizík** – je pridelenie vlastníctva rizika inej strane (subjektu). Príkladom môže byť keď manažér navažuje partnerský vzťah s partnerom projektu.
3. **Posilnenie rizika** – je zmena veľkosti príležitosti, ktoré dosiahneme pomocou identifikácie a maximalizácie kľúčových faktorov pozitívneho rizika.
4. **Prijatie rizika** – platí i pre pozitívne riziká, a to v prípade, že projektový tím nemôže k vzniku rizika prispieť žiadnym aktívnym opatrením, alebo sa skratka rozhodne také opatrenia neurobiť.

Hlavným výstupom z plánovania reakcie na riziká sú zmluvné ujednávania, ktoré súvisia s rizikami a aktualizovaná verzia plánu riadenia projektov a registrov rizík. Súčasťou stratégie na reakciu rizík bývajú často informácie o reziduálnych a sekundárnych rizikách a tak isto mimoriadne vyššie popísané plány a rezervy [5].

4.8 Sledovanie a kontrola rizík

Do činnosti monitorovania alebo sledovania a kontroly rizík riadime prevedenie taktových procesov riadenia rizík, ktoré reagujú na konkrétne rizikové udalosti. Povedenie týchto procesov riadenia rizík znamená zabezpečiť, že udržiavanie podvedomia o rizikách bude celý projektový tím sledovať ako priebežnú aktivitu a bude ju robiť po celú dobu riešenia projektu.

Riadenia rizík v projektoch rozhodne nekončí u začiatkovej analýzy rizík. Riziká, ktoré sme takto identifikovali, sa nemusia vôbec prejavíť alebo sa pravdepodobnosť ich vzniku alebo hodnota straty môže zmenšiť, prípadne celá zmiznúť. Podobne v behu riešenia projektov identifikujeme aj nové riziká, ktoré musia prejsť takým istým procesom ako riziká, identifikované pri začiatkovej analýze rizík. Keď dôjde k zmenám v relatívnom vystavení projektu jednotlivým rizikám, môže byť niekedy potreba zmeniť si aj rozdelenie zdrojov vyhradených na riadenia rizík.

Pri vykonávaní jednotlivých plánov riadenia rizík musíme sledovať riziká podľa definovaných míľnikov a prijímať rozhodnutia ohľadne rizík a stratégií reakcie na nich. Keď sa určitá stratégia ukáže byť neúčinnou, keď implementujeme plánovanú mimoriadnu aktivitu alebo odstránime zo zoznamu riziko, ktoré už prestalo existovať musíme stratégiu rizika zmeniť. Projektové tímy siahajú často aj k rôznym **náhradným riešeniam** (workarounds) – je to neplánovaná reakcia na rizikovú udalosť, ktorá nie je pokrytá vhodným mimoriadnym plánom.

Výstupom z popisovaného procesu sú požadované zmeny, doporučené nápravné opatrenia, preventívne opatrenia a aktualizovaná verzia registra rizík, plánu riadenia projektov a organizačných aktivít organizácie ako sú napríklad poučené informácie, ktoré využijeme pri riešení nových a budúcich projektov [1].

4.9 Softwarová podpora riadenia rizík v projektoch

V rôznych procesoch riadenia rizík môžeme využívať možnosti najrôznejších softwarových nástrojov. Vo väčšine organizácií prebieha pomocou softwarových nástrojov vytváranie, aktualizácia, a distribúcia informácií z registra rizík. Ten býva často podobný jednoduchému súboru v aplikácii Microsoft Word alebo Excel, ale môže byť súčasťou zložitejšej popracovanej databázy.

Pri sledovaní a kvantifikácii rizík, príprave rôznych grafov a diagramov nám pri prevedení analýzy citlivosti pomáhajú tabuľkové procesory.

Ďalšou oblasťou pre uplatnenie softwarovej podpory je vytváranie rozhodovacích stromov a odhadu očakávanej peňažnej hodnoty. Vyspelejšie softwarové nástroje pre riadenie rizík, ako je napríklad stimulačný software pre metódu Monte Carlo, nám pomáhajú s vývojom modelu, s analýzou rizík a stanovením reakcie na ne prostredníctvom simulovaných výpočtov.

Aj keď nové softwarové nástroje značne uľahčujú zložitejšie typy analýzy rizík, nesmú sa projektové tímy pri riadení rizík v projektoch na ich možnosti slepo spoliehať. Keď určité riziko neidentifikujeme, nemôžeme ho ani riadiť, pretože dobrú identifikáciu rizika môžu previesť len dostatočne skúsení a inteligentní ľudia. Dosť práce je tak isto vytvoriť si a následne implementovať dobrú stratégiu pre reakcie na riziká.

Software by sme mali používať len ako pomocný nástroj, čo nám pri riadení rizík v projektoch uľahčuje prijatie tých správnych rozhodnutí, a v prípade zlyhaní projektu nesmieme na software zvaliť všetky prípadné problémy.

5 Analýza a návrh

Cieľom tejto kapitoly je popísať analýzu a návrh systému pre podporu managementu rizík v IT projektoch. Vo fáze analýzy je definovaný problém, ktorý vychádza z úvodných jednaní so zákazníkom. Patria sem neformálne špecifikácie, najdôležitejšie funkcie systému a modely navrhovaného systému. Tieto štúdiá je možné chápať ako predbežné, ktoré už dávajú celkom stabilnú predstavu o funkčnosti systému. Z neformálnych špecifikácií vychádzajú potom požiadavky na systém, ktoré dopĺňajú diagramy prípadu použitia (Use Case Diagrams). Tieto diagramy nám ukazujú chovanie systému bez ohľadu na vnútornú štruktúru.

V tejto fáze som urobil podrobnú analýzu problematiky riadenia rizík v projektoch, ktoré sú typické pre oblasť IT. Keďže veľa softwarových podpor na tento problém riadenia neexistuje, alebo sú špecifické na konkrétnu oblasť, musel som absolvovať veľa stretávok a náročných konzultácií s mojou konzultantkou ohľadne dôkladnej analýzy problematiky. Z načerpaných skúseností som vytvoril prehľad možných funkcií systému, ktoré sú podľa mňa dôležité, ale v žiadnom prípade nie zložité pre používateľov.

Vo fáze návrhu som sa musel niekoľkokrát vrátiť späť do etapy analýzy a previesť nemalé úpravy. Podobne tomu bolo aj vo fáze implementácie. Dôvodom môže byť aj nepríliš konkrétna špecifikácia a špecifickosť okruhu tematiky, pretože riadenie rizík je ešte vo fáze manažérskeho vývoja a je špecifická pre každú oblasť zvlášť.

Fáza návrhu začína tam, kde končí analýza požiadaviek. Táto fáza obsahuje popis štruktúry softwarového systému, ktorý má byť implementovaný, dáta ktoré sú súčasťou systému, rozhranie komponentov a používateľské rozhranie. V prípade informačných systémov popis dát predstavuje návrh štruktúry databázy.

Nasledujúce podkapitoly popisujú jednotlivé fázy analýzy a návrhu, ktoré sú očakávané funkcie systému zistené vo fáze analýzy, samotný diagram prípadu použitia, štruktúra databázy s diagramom ERD a návrh grafického používateľského rozhrania.

5.1 Neformálna špecifikácia požiadaviek

Cieľom projektu je programová podpora pre riadenie rizík v IT projektoch. Je to vlastne určitý typ, podmnožina projektového riadenia, ktorá sa týka všetkých oblastí a je veľmi dôležitou súčasťou každého projektu. Výsledná aplikácia by mala byť jednoduchá s intuitívnym grafickým používateľským rozhraním. Mala by umožňovať prehľadným spôsobom riadiť riziká v projektoch a poskytovať grafické výstupy pre manažérske rozhodnutia.

Systém nie je možné porovnávať s komerčnými systémami, pretože neexistuje žiadny poznaný všeobecný systém na túto tematiku. Preto bola samotná analýza, návrh a špecifikácia požiadaviek na systém veľmi ťažkou súčasťou celého projektu, lebo inšpirácia z iných komerčných systémov nebola možná. Aplikácia by mala zvládnuť projekty rôzneho rozsahu s rôznymi kategóriami a typmi rizík. Riziká je treba evidovať na úrovni procesov, ktoré obsahujú jednotlivé projekty. Výsledný systém by mal mať podobu oknovej aplikácie, ktorá bude bez problémov funkčná na bežných kancelárskych počítačoch s nainštalovaným MS SQL serverom. Aplikácia bude využívať klasickú trojvrstvovú architektúru, ktorá bude popísaná nižšie v ďalšej podkapitole návrhu.

5.1.1 Očakávané funkcie systému

Prvou dôležitou vlastnosťou systému by malo byť dostatočné zabezpečenie a návrh používateľských oprávnení podľa ich funkcionality v systéme. Každý používateľ by mal mať prístup len k tým funkciám, ktoré skutočne potrebuje a naopak by nemal mať možnosť previesť v systéme činnosti, ktoré mu jeho rola na projektoch, procesoch a rizikách nepovoľuje. Prvou funkciou by teda malo byť bezpečné prihlásenie používateľa do systému.

Používatelia systému

Práca používateľa v systéme by mala byť rozdelená do troch úrovní prístupu, ktorá im bude pridelená pri registrácii. Do systému teda bude možné prísť ako správca systému, projektový manažér a pracovník, čo zahŕňa všetky ostatné funkcie na projekte, ktoré v rámci riadenia rizík nie sú dôležité.

Používatelia systému s právom **pracovníka** by mali mať v systéme najmenej privilégií. Môžu sa pozrieť na jednotlivé rozpracované projekty a procesy, ale nemajú právo na zakladanie alebo zmenu projektov a procesov. O týchto projektoch a procesoch sa môžu vyexportovať, vytlačiť informácie, reporty a štatistické súhrny. Používatelia s právom pracovníka budú pridelení k projektom ako osoby, ktoré sa na nich zúčastnia a to im umožní identifikovať riziká a prideliť ich k úlohám. Kým nebudú vlastníkom nejakého rizika, nebudú môcť robiť žiadne iné úpravy s rizikami, okrem štatistických súhrnov a exportov. Akurát je pracovník zodpovedný za riziko a môže toto riziko v plnom rozsahu spravovať. Používatelia s právom pracovníka samozrejme budú mať v systéme umožnené zmeniť si vlastné údaje zadane pri registrácii. To, že ktorý pracovník je priradený ku ktorému projektu budú určovať projektoví manažéri pri zakladaní projektov.

Právo **projektového manažéra** zahŕňa všetky možnosti, ktoré bude mať používateľ s právom pracovníka. Okrem toho by mal umožňovať zakladanie projektov a procesov a previesť ich úplnú správu. Projektový manažér, ktorý nebude zakladateľom projektu, bude môcť disponovať s menšími právami, ktoré budú skoro rovnaké ako u jednoduchého pracovníka. Bude môcť previesť exporty štatistík, vytlačiť grafovú analýzu, alebo exportovať do .pdf súborov. Projektový manažér tiež bude môcť pridelovať aj zodpovednú osobu (vlastníka) za riziko, ktoré bolo registrované k procesom. Ešte pred samotným začiatkom projektu každému manažérovi musí byť k dispozícii možnosť výpočtu očakávanej peňažnej hodnoty s grafickým výstupom a hodnotením.

Posledným používateľom systému je **administrátor**. Má najvyššiu úroveň prístupu. Vlastní všetky oprávnenia manažéra, pracovníka a má úplný prístup k všetkým projektom, procesom a rizikám, ktoré sú v systéme evidované. Bude ako jediný oprávnený môcť registrovať nového používateľa systému a prideliť mu potrebné právo, s ktorým bude figurovať v systéme. S pohľadom projektového riadenia s právom administrátora, budú v systéme evidovaní najvyšší činitelia v rámci organizácie napr. riaditeľ, konateľ alebo nimi poverená osoba.

Projekty a procesy

Z jednotlivých rolí je zrejmé, že projekty budú vytvárať projektoví manažéri a prevedú aj ich celú správu. Procesy budú priradené už k jednotlivým projektom. Definícia procesu v tomto projekte zahŕňa úlohu, udalosť alebo správu, ktorá bude priradená ku konkrétnemu vytvorenému projektu. Svojou podstatou sú tieto procesy počas života projektu rovnaké s obecnými procesmi.

Ešte pred samotným vytvorením konkrétneho projektu, systém bude umožňovať previesť **výpočet očakávanej peňažnej hodnoty**, podľa čoho projektový manažér bude môcť rozhodovať o výbere medzi projektmi. Systém by mal naraz porovnať dva projekty podľa vstupných údajov.

Výstupom by mal byť stromový graf rozhodnutia a porovnávanie podľa výsledkov s exportom do .pdf a možnosťou tlače.

Po vytvorení a priradení pracovníkov k projektu bude možné pridať aj procesy – úlohy, pre ktoré potom budú evidované i riziká. Každý proces môže byť rozdelený na viac podprocesov. Každý proces teda môže mať definovanú **väzbu na nadriadený proces**. Riziká by sa mali byť definované priamo k jednotlivým podprocesom, aby bola vytvorená hierarchia a prehľadnosť určená k jednotlivým úrovňam.

Projektový manažér, ktorý projekt vytvoril by mal byť automaticky priradený k tomuto projektu a mal by byť jediným hlavným manažérom fázy. Len on bude mať právo pridať **nové procesy**, zmeniť informácie o projekte, procesoch a prípadne zmazanie nepotrebných častí. Mal by mať celkovú kontrolu nad svojimi projektmi a priradenými pracovníkmi. Keď sa pracovník prihlási do systému, najprv by mal mať k dispozícii tie projekty, na ktoré je priradený, ale musí mať možnosť výberu aj iného projektu, na ktorých momentálne nepracuje.

Pretože tento systém nie je hlavne zameraný na riadenie projektov a úloh, ale na riadenie rizík, ktoré patria k týmto častiam. Samotné riadenie projektov a procesov preto nemusí byť rozsiahlo rozpracované, len ak je to nevyhnutné pre pracovanie s rizikami. Jedinou nevyhnutnou podmienkou je, že riziká musia byť evidované k jednotlivým projektovým úlohám hierarchicky a nie na celkový projekt všeobecne. Ako hlavný dôvod tejto skutočnosti vidím v tom, že riadiť riziká nie je postačujúce na úspešné dokončenie IT projektov na úrovni celého projektu. Tieto riziká si musia priradovať až k najnižšiemu možnému procesu pre ich efektívne a prehľadné využitie.

Riziká

Riziká sú najdôležitejšou časťou celého programu, preto musia byť aj najviac rozpracované. Ako bolo zmienené, riziká musia byť pridelené k procesom so všetkými svojimi vlastnosťami, ale u niektorých prípadov ich štatistické súhrny budú postačujúce na celý projekt.

Riziká budú môcť do systému pridávať všetci používatelia, ktorí sú priradení na daný projekt, v ktorom identifikujeme riziko. Po identifikácii už riziko bude spravovať len projektový manažér konkrétneho projektu, alebo priradený vlastník rizika. Týmto vlastníkom môže byť hociktorý projektu priradený pracovník. Títo aktéri budú môcť potom zmeniť vlastnosti rizika, prípadne priradiť iného vlastníka.

Pridať riziko sa bude zadávať pomocou sprievodcu. Na začiatku bude možné vybrať riziko **z centrálnej databázy**, kde budú vymenované všetky riziká uložené v systéme. Keď už budú riziká priradené k procesom projektu, potom sa priradia aj všetky jeho meniace atribúty. Tieto atribúty potom nebudú uložené v centrálnej databáze, lebo sa menia u každého rizika. Samozrejme riziko nemusí byť vybrané z centrálnej databázy, ale môže sa nadefinovať od začiatku samostatne. Pri pridaní bude potrebné zadať aj údaje ako napr. príčina vzniku rizika, spúšťače, kategória, typ ...atď. Medzi najdôležitejšími atribútmi bude **pravdepodobnosť** a **dopad** vzniku rizikovej udalosti. Pre tieto hodnoty bude treba nadefinovať aj **tabuľku hodnotenia**, ktorá môže byť pre každé riziko iná. Z týchto atribút potom bude vypočítané dané riziko, ktoré bude udávať závažnosť rizikovej udalosti. Pre každé riziko by mala byť možnosť **pridania reakcie** v prípade vzniku rizikovej udalosti. Túto reakciu je možné pridať pri zadaní samotného rizika, alebo je ju možné pridať pri neskorších úpravách.

Kategórie typov a reakcií na riziká bude spravovať projektový manažér daného projektu, ktorý by mal mať aj možnosť priradiť existujúce typy a reakcie pridané u iných rizík.

V rámci implementácie bude treba vyriešiť aj **pridávanie existujúcich súborov** k rizikám, ktoré budú napríklad zápisnice konferencií, kde bolo riziko zistené, prípadne tabuľky vytvorené

v programu Excel. Systém bude musieť akceptovať aspoň súbory s príponami .doc, .xls, .txt, .pdf a umožňovať priamu prácu s týmito súborami z programu.

Systém bude umožňovať u každého rizika evidovať aj **časový graf pravdepodobnosti a dopadu**, ktorý bude zobrazovať hodnoty týchto dvoch veličín v jednotlivých mesiacoch počas trvania projektu. V prvom rade si bude treba dávať pozor pri implementácii, aby hodnoty týchto veličín boli monitorované a vo vhodnom časovom okamihu uložené. Tieto hodnoty potom bude potrebné graficky prehľadne zobrazovať a umožňovať funkciu tlače.

Monitorovanie rizika bude prebiehať aj formou **karty rizika**. Karta rizika slúži pre základnú evidenciu informácií o pridelenom riziku pre výstup. Manažérovi bude slúžiť ako upresňujúci podklad k pravidelnej správe o riziku. Karta rizika bude musieť obsahovať aj všetky dôležité sledované údaje o riziku a bude pravidelne aktualizovaná. Môže byť v časových intervaloch vyexportovaná do PDF súboru a priložená k riziku ako referencia. Spolu s časovým grafom a mapou rizika i karta rizika bude môcť byť súčasťou súhrnnej správy o rizikách.

Mapa rizika je vlastne graf pravdepodobnosti a dopadu všetkých rizík uložených v projekte. Táto mapa musí byť postupne aktualizovaná a systém musí umožniť jej export do súboru aj na tlačiareň. Z implementačného hľadiska sa musí klásť veľký dôraz na jeho správne zobrazenie, pretože musia byť pravdepodobnosti a dopady rizík podľa ich hodnotenia v mape zobrazené na jednotné intervaly.

Každé riziko, ktoré bude identifikované a pridané do systému, bude možné **vyexportovať do XML** súboru so všetkými priradenými aktérmi a udalosťami. Tento XML súbor sa potom bude môcť meniť v iných programoch a prípadne nainportovať späť do iného projektu, procesu. Program by mal obsahovať funkcie pre export a na tlač najdôležitejších štatistík do .pdf súboru, alebo na tlačiareň.

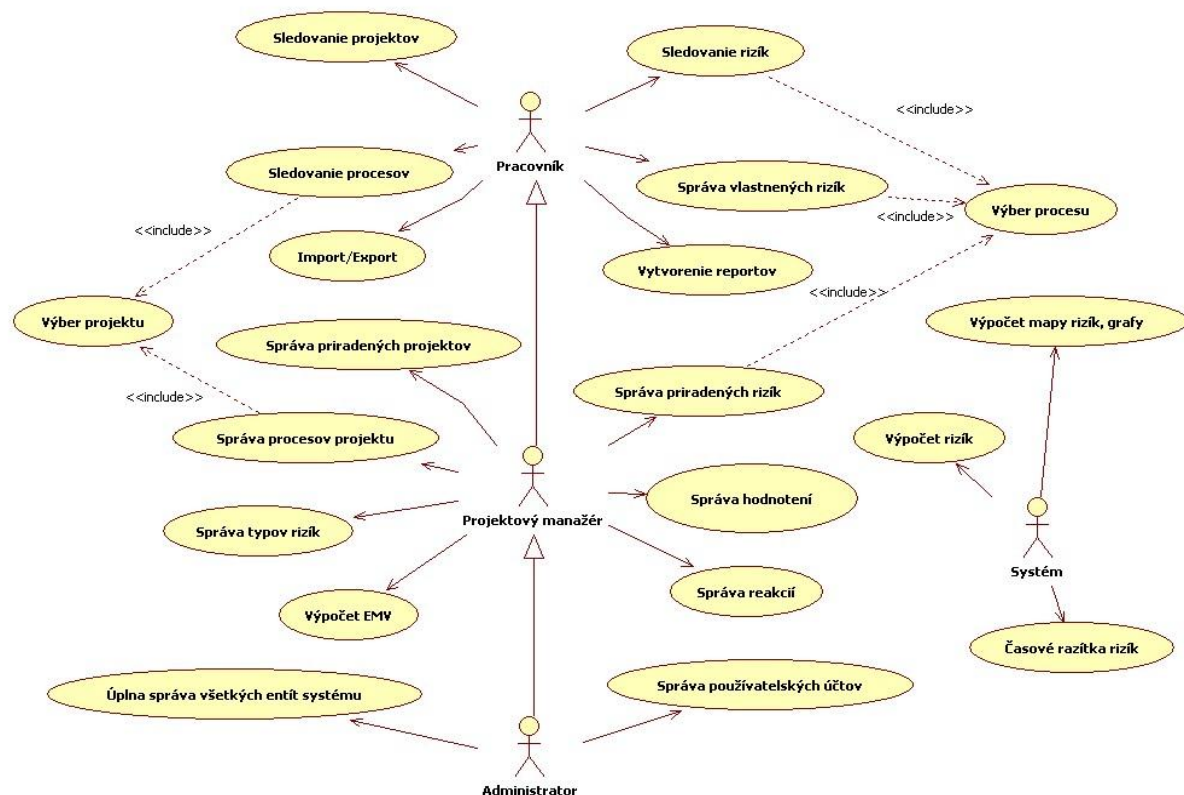
5.2 Diagram prípadu použitia

Na modelovanie diagramu prípadu použitia som použil zjednotený modelovací jazyk UML (Unified Modeling Language). Tento jazyk sa používa najčastejšie pri modelovaní informačných systémov a budem ho používať aj pri modelovaní návrhu databázy. Diagram prípadu použitia (obrázok 5.1) vychádza z neformálnej špecifikácie a predstavuje chovanie systému z pohľadu používateľa. Je to diagram, ktorý umožňuje pochopiť podstatu problému a aj spôsob jeho riešenia. Preto je dôležité, aby diagram reprezentoval systém na nízkej úrovni podrobnosti, aby bol prehľadný aj pre zadávateľa aj pre vývojára.

Pretože model zachycuje funkcionality systému z pohľadu používateľa, spôsob zápisu požiadaviek musí byť teda čitateľný hlavne pre neho. Z tohto vyplýva, že detailnejší návrh, ktorý by musel byť rozpracovaný do vyššej úrovni podrobnosti by bol zbytočný a preto je žiadajúca jednoduchosť. Prípad použitia má svoj:

- názov,
- jedinečného identifikátora,
- špecifikáciu.

Na modely, ktorý je znázornený na obrázku 5.1 je vidieť, že k systému pristupujú tri typy používateľov, ktorí sú definovaní v neformálnej špecifikácii. Títo používatelia budú pristupovať k systému a budú využívať jeho funkcie.



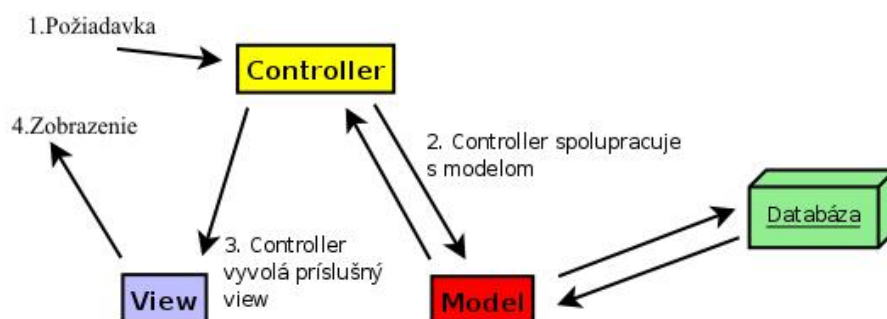
Obrázok 5.1: Diagram prípadu použitia systému

Jediným nefyzickým aktérom je samotný systém, ktorý má na starosti postupné spracovanie a aktualizovanie grafov, výpočet mapy rizík a závažnosti rizika. Systém bude mať na starosti aj sledovanie a ukladanie časových hodnôt v jednotlivých etapách.

5.3 Návrh architektúry

Návrh architektúry celého systému určuje základnú konštrukciu vyvíjaného produktu. Je to vlastne postup ako navrhnuť základnú kostru a aspekty aplikácie tak, aby spĺňal funkčné a mimofunkčné požiadavky, aby bol vývoj systému priamočiary a prevádzkovanie bezproblémové. Rozdelenie architektúry sa stáva dôležitou súčasťou návrhu hlavne pri veľkých systémoch.

Práve z tohto dôvodu som si vybral pre návrh mojej architektúry architektonický vzor MVC (Model – view - controller). Model MVC nám aplikáciu rozdelí na niekoľko vrstiev, čo je vidieť na obrázku 5.2.



Obrázok 5.2: Rozdelenie vrstiev architektúry MVC[13]

Všeobecne povedané, rozdelenie vrstiev – vytváranie aplikácií s využitím architektúry MVC vyžaduje vytvorenie troch komponentov, medzi ktoré patria:

Model - vrstva zodpovedná za spravovanie stavu aplikácie. V niektorých prípadoch stav trvá len krátko - len počas komunikácie s používateľom, inokedy je trvalý, ukladaný mimo aplikácie - najčastejšie v databáze. Model poskytuje prístup k dátam a umožňuje ich ukladanie a aktualizáciu. Model nie je zodpovedný len za samotné dáta, obsahuje tiež metódy na aplikovanie pravidiel a podmienok viažucich sa na údaje. Napríklad ak naša aplikácia má na starosti evidenciu rizík, úlohou modelu je vypočítať podľa pravdepodobnosti a dopadu závažnosť rizika atď. Dáva to zmysel, lebo model sa stará o samotnú konzistenciu správnosti dát.

View – teda „pohľad“ sa stará o reprezentáciu a formátovanie údajov poskytnutých modelom. K jednému modelu môže a zvyčajne aj existuje viacero pohľadov. Napríklad pre model systému riadenia rizík môže existovať pohľad pre projektového manažéra, rovnako pre ostatných pracovníkov, hoci obaja používajú ten istý model, ikeď pohľad a aj funkcionality je rôzna.

Controller – teda radič je vrstva zodpovedná za riadenie aplikácie, prijíma vstup od používateľa, vyžiada si údaje od modelu a pošle správny view späť používateľovi.

Výhoda MVC architektúry spočíva v tom, že je to ustálená architektúra, ktorá pretvorí aplikáciu na udržateľný, modulárny a rýchlo sa vyvíjajúci balík. Nové vlastnosti sú pridávané jednoducho a rýchlo. Modulárny a oddelený návrh taktiež umožňuje návrhárom a vývojárom pracovať súčasne a zahŕňa taktiež možnosť rýchleho prototypovania. Separácia taktiež umožňuje vývojárom robiť zmeny v jednej časti aplikácie bez ovplyvnenia zvyšných častí [13].

5.4 Návrh databázy

Návrh databázy vychádza z fázy analýzy a definuje model, ktorý určuje ako budú dáta uložené v databáze. Bez znalostí požiadaviek na systém nie je možné samotný návrh databázy previesť.

Snažil som sa systém a databázu navrhnuť tak, aby v prípade rozširovania nebola potrebná žiadna radikálna zmena. To by malo zaručiť dobré možnosti pre ďalší rozvoj systému do budúcnosti.

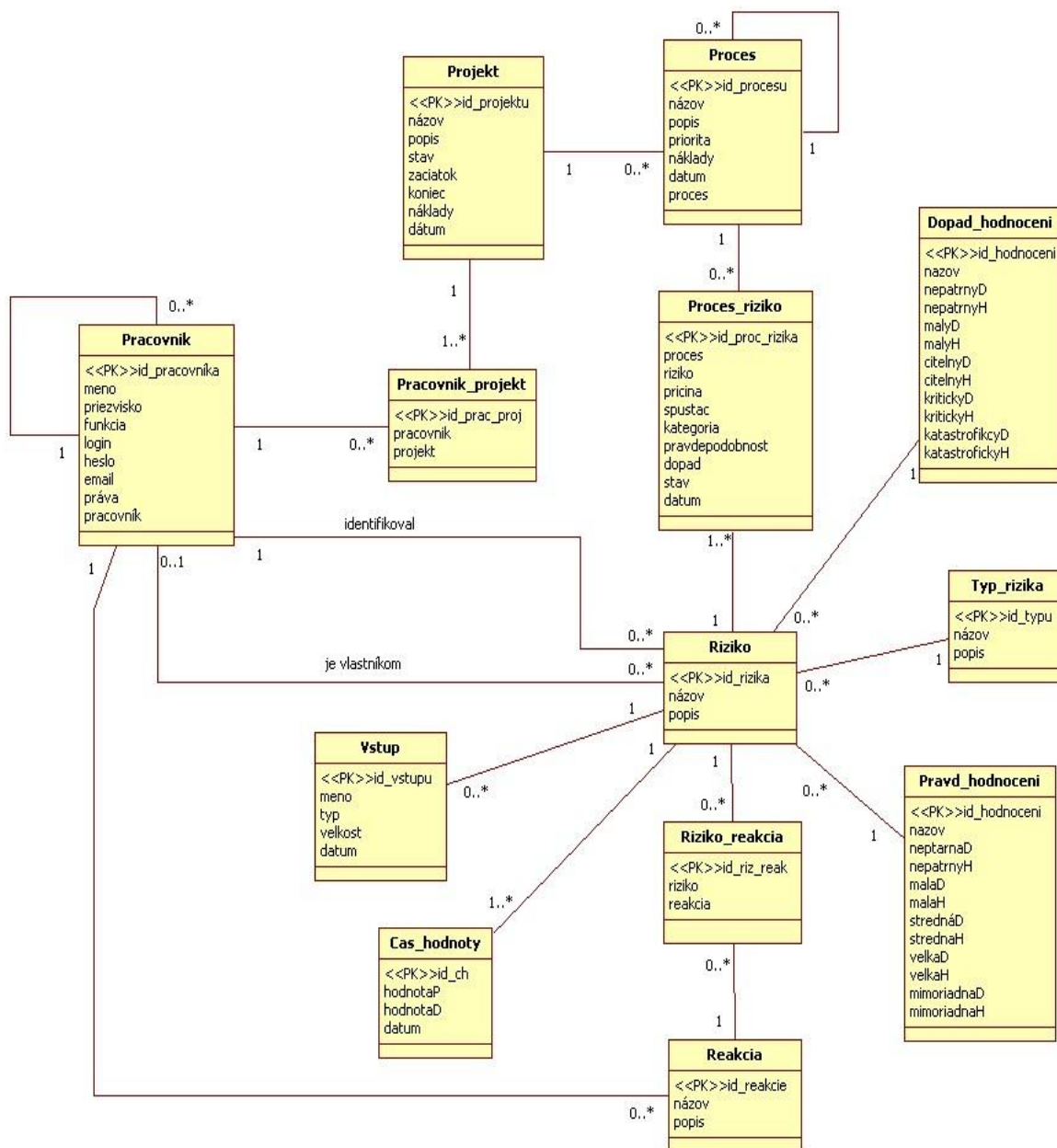
Pre modelovanie návrhu databázy som používal relačný model s tabuľkovou štruktúrou. Táto forma modelovania sa nazýva modelovanie pomocou ER diagramov. ER diagram je grafická reprezentácia entít, vzťahov a obmedzení, ktoré tvoria daný návrh. Tak isto ako u iných grafických návrhových metodológií, poskytuje grafický súhrn. Je to veľmi užitočné pre návrhára, nielen pre validáciu správnosti návrhu, ale i pre diskusiu s kolegami a programátormi, ktorým je treba problém vysvetliť. Neexistujú obecné platné konvencie pre nakreslenie E-R diagramov a preto môžu existovať rozdiely a rôzne typy E-R diagramov. E-R diagram obsahuje entity, atribúty, vzťahy:

- **entita** – vec reálneho sveta (objekt) rozlíšiteľný od iných objektov,
- **atribút** – vlastnosť entity, ktorá nás v kontextu daného problému zaujíma,
- **vzťah** – asociácia medzi niekoľkými entitami.

Pri vytvorení E-R diagramu je treba sa rozhodnúť, ktoré entity, vzťahy a obmedzenia použijeme pre modelovanie fyzického systému. Pri jednoduchom príklade môže toto rozhodnutie byť jednoduché, ale pre praktické informačné systémy ide čiastočne o umenie a skúsenosti.

Dôležitou výhodou tejto metodológie je, že sa návrhár môže sústrediť na presné modelovanie fyzického systému bez toho, aby sa zaťažoval efektivitou cieľovej databázy.

Hlavnou entitou celého systému bude *Riziko*. Tieto riziká budú priradené k procesom pomocou tabuľky *Proces_riziko*. Jedno riziko môže patriť do viac procesov, ale minimálne do jedného musí, preto je násobnosť vzťahu týchto entít 1-1.*. Spojovacia tabuľka, ktorá je medzi týmito entitami, obsahuje atribúty, ktoré sa budú meniť pri priradovaní rizika ku konkrétnemu procesu. Atribúty tabuľky riziko budú použité len u centrálnej databázy rizík. Samotná entita *Proces* je vo vzťahu 0..*-1 s entitou projekt. Teda do jedného projektu môže patriť ľubovoľný počet procesov alebo aj nula, ale pri vytvorení nejakého procesu ho musíme priradiť do projektu. Znamená to,



že proces bez projektu v systéme nebude dovolený. Procesy môžu byť ľubovoľne zanorené, preto je na obrázku 5.3 uvedený u entity Proces aj vzťah sám na seba s kardinalitou 1-0..*.

K entite *Projekt*, ktorý môže obsahovať ľubovoľný počet procesov, alebo ani jeden (značí kardinalita 1-0..*) budeme môcť priradovať aj ľubovoľný počet pracovníkov. Keďže pracovník zaregistrovaný v systéme, môže byť priradený aj u iných projektov vznikol by vzťah *-*, ktorý by implikoval, že databáze by nevyhovela normálnym formám, teda musí byť medzi nimi zase spojovacia tabuľka (*Pracovnik_projekt*). V entite *Pracovnik* je atribút právo, podľa čoho budeme identifikovať aké právo bude mať pracovník prístupujúci k systému. Pracovníka musíme priradiť k riziku aj ako vlastníka aj ako identifikujúceho. K týmto vzťahom pripadajú kardinality 1-0..* a 0..1-0..*. Pracovníka priradíme i samého k sebe pre identifikáciu nadriadeného a tak isto k entite *Reakcia*, ktorá bude identifikovať pracovníka, ktorý zadal reakciu na riziko.

K riziku bude priradený aj typ rizika reprezentovaný entitou *Typ_rizika*, ktorý môže byť priradený aj k viacerým rizikám. Tak isto budú priradené aj tabuľky hodnotenia pravdepodobnosti (*Pravd_hodnoceni*) a dopadu (*Dopad_hodnoceni*). U týchto entít kardinalita bude 0..*-1. Aby bolo možné u každého rizika uchovávať aj priebeh hodnôt v čase, treba v databáze vytvoriť tabuľky pre tieto časové hodnoty. Budú uložené v tabuľke *Cas_hodnoty*, kde k riziku bude vždy priradená aspoň jedna hodnota tejto entity. Pretože systém bude umožňovať k rizikám priradovať aj externé súbory, informácie o týchto súboroch treba tak isto uložiť v databáze. Tieto údaje budú uložené v tabuľke pod názvom *Vstup*. Aj v tejto tabuľke, tak isto ako u každej v hodnotách atribút uložíme tie najdôležitejšie údaje, aby databázových operácií nebolo zbytočne veľa. Poslednou entitou priradenou k riziku, ktorú je vidieť na obrázku 5.3 je tabuľka *Reakcia*. Tak isto ako v predchádzajúcich prípadoch, keď by vzťah implikoval kardinalitu *-* je potrebné vložiť spojovaciu tabuľku, v našom prípade *Riziko_rekacia*. Systém teda bude dovoliť priradenie ku každému riziku nekonečný počet reakcií. Tieto reakcie, ktoré sú v systéme evidované, potom bude možné priradiť i k viacerým rizikám. Entity znázornené v ER diagrame neobsahujú atribúty cudzích kľúčov do iných entít, pretože je ich možné odvodiť zo vzťahov medzi nimi.

Tak ako vo fáze analýzy boli prevedené časté zmeny v požiadavkách, aj fáza návrhu bola zaťažovaná častým a neefektívnym upravovaním dátového modelu. Mnoho zmien bolo nutné previesť behom zoznamovania sa s dátovým modelom vyvíjaného informačného systému a jeho možnosťami. Z dôvodu potrebnej kompatibility pre neskorší rozvoj systému, som bol nútený vytvoriť si najuniverzálnejší návrh dátového modelu. Časté doplňovanie a úpravy modelu boli zapríčinené s nedostatočne detailnou špecifikáciou požiadaviek. Aj pri implementácii som sa musel niekoľkokrát vrátiť späť na úroveň dátového modelu a upravovať ho.

5.5 Návrh grafického rozhrania

Návrh grafického používateľského rozhrania definuje prostredie, ktoré je pre používateľov dôležitou súčasťou celého systému. Používateľ systému celý čas bude s programom komunikovať pomocou tohto rozhrania, preto je treba rozhranie vyvíjať podľa ich predstáv, skúseností a podľa ich návykov. Veľmi užitočný môže byť návrh pomocou vývojového prostredia pre programátorov, z ktorého bude prehľadné a jasné, čo sa bude používateľovi v jeho súčasnom stave zobrazovať a aké bude mať ďalšie možnosti.

Vzhľadom k tomu, že vyvíjaný systém bude implementovaný, ako klasická oknová aplikácia operačných systémov Windows, používateľské rozhranie som sa rozhodol vytvoriť pomocou návrhára Visual Studio. Ten umožňuje rýchle a pohodlné vytvorenie všetkých potrebných ovládacích

prvkov s výberom zo zoznamu dostupných komponentov a umiestnením na požadovaný formulár. Rozmiestniť prvky je možné umiestňovaním týchto prvkov do vhodných kontajnerov.

Po spustení aplikácie vyjde prihlasovacie okienko pre bezpečnú autentizáciu používateľa. Po prihlásení bude zobrazené okno výberu projektu, s ktorým bude možné ďalej pracovať. Samozrejme musí byť aj možnosť založenia nového projektu pre projektových manažérov a správa používateľov pre administrátorov. Z tohto okna by mal byť dostupný aj formulár možnosti výpočtu očakávanej peňažnej hodnoty. Tento formulár bude obsahovať vstupné pole, ktoré používateľ musí vyplniť pre korektný výpočet tejto peňažnej hodnoty. Výsledok výpočtu bude zobrazený graficky stromovou štruktúrou.

Hlavné okno aplikácie bude už ukazovať prácu s konkrétnym projektom a jeho rizikami. Bude obsahovať hlavné menu, z ktorého je možno vyvolať všetky funkcie. Menu bude členené do položiek, ktoré logicky zoskupujú funkcie podľa ich účelov. Pod hlavným menu sa bude nachádzať nástrojová lišta, ktorá bude umožňovať spúšťanie často používaných funkcií jednoducho, kliknutím na určitú ikonu. Obrázky ikon boli vytvorené tak, aby znázorňovali čo najvýstižnejšie danú funkciu. Pre štandardné funkcie, ako je otvorenie, vytvorenie a uloženie, boli použité obrázky známych aplikácií systému Windows. Používatelia sú už na to zvyknutí a prípadné používanie odlišných obrázkov by im bolo ťažšie na zvyknutie.

Pod nástrojovou lištou bude okno rozdelené na dve stránky. Na ľavej strane prvej stránky bude obsadená stromová štruktúra projektu s procesmi, kým na pravej strane bude výpis rizík s informáciami. Rizika by mali byť zobrazené podľa výberu procesu, alebo celého projektu zo stromovej štruktúry. Druhá stránka hlavného okna aplikácie by mala obsahovať grafické zobrazenie mapy rizík s potrebnými nastaveniami a popismi vlastností. Tieto okná by mali obsahovať aj rôzne grafické tlačidlá pre navigáciu a prácu s rizikami. Nad výpisom tabuľky rizík by mal byť umiestnený aj jednoduchý filter, ktorý by umožňoval filtrovať riziká podľa kategórií. Pod tabuľkou rizík budú aj grafy, ktoré budú zobrazovať štatistiku kategórií a závažností už identifikovaných rizík.

Pridávanie rizika bude realizované pomocou sprievodcu pridávania, ktorý by postupne žiadal používateľa krok za krokom na vyplnenie všetkých údajov zoskupených podľa ich účelov. V rámci pridania, užívateľ by mal možnosť vyvolať aj formuláre na priradenie iných typov, reakcií a priradenie súborov. Všetky tieto dialógy by mali byť dostupné aj z hlavného okna aplikácie.

Podrobné informácie o rizikách budú vypísané do okna, obsahom viac stránok. Stránky budú kategorizované podľa podobnosti vlastností, obdobne ako u sprievodcu pridania rizika. Z týchto okien by mal byť dostupný aj odkaz na export do XML súboru a aj grafické zobrazenie karty rizika. Tak isto ako z hlavného okna, z okna podrobností rizika bude aj odkaz na grafické zobrazenie časových hodnôt. Všetky grafické výstupy týkajúce sa rizík budú mať možnosť exportu a tlače.

Ďalšie funkčné jednotky aplikácie ako pridanie nového procesu, profil používateľa, zmena údajov, import – export a tlač by mali byť realizované podobnými formulármi. Celkový vzhľad všetkých okien a formulárov by mal byť podobný, aby si používateľ zvykol na jednotný koncept a ľahko sa v systéme orientoval.

Táto kapitola popísala analýzu a návrh aplikácie pre riadenie rizík v IT. Na začiatku kapitoly boli popísané neformálne špecifikácie a očakávané funkcie systému. Tieto funkcie boli z pohľadu používateľa graficky zobrazené v diagrame prípadu použitia. Na analýzu priamo nadviazal aj návrh systému. Návrh bolo treba urobiť zvlášť na architektúru, databázu s diagramom ERD a na grafické používateľské rozhranie. Tieto návrhy sú základným kameňom pri implementácii a prípadné chyby by mohli spôsobiť aj nefunkčnosť celého systému. Keď je úspešne dokončený návrh a analýza, je možné pristúpiť k samotnej implementácii systému, čo popisuje nasledujúca kapitola práce.

6 Implementácia

Fáza implementácie systému vychádza z návrhu a zaoberá sa samotným vývojom aplikácie, teda väčšinou s programovaním. Pritom to neznamena len kódovanie návrhu do príkazov programovacieho jazyka, ale môže obsahovať aj návrh algoritmov a zložitých SQL dotazov, ktoré sú intelektuálne náročné a nie sú vždy súčasťou návrhu systému. Moderné programovanie je pritom aj založené na využití existujúcich či dostupných komponentov. To vyžaduje od programátora nielen znalosť programovacieho jazyka, ale i znalosť či schopnosť použiteľné komponenty vyhľadať a používať. Programovací jazyk C# s technológiou .NET vo vývojovom prostredí Visual Studio umožňuje takéto komponenty používať vo viacerých prípadoch. Práve tento programovací jazyk spolu s technológiou a vývojovým prostredím bol využívaný aj pri implementácii tohto systému.

V ďalších podkapitolách sa budem venovať popisu týchto použitých technológií pri realizácii výslednej aplikácie. Ďalej budú popísané aj implementované funkcie a ich vlastnosti spolu s problémami, na ktoré som narazil pri implementácii.

6.1 Implementačné nástroje

Aplikáciu som sa rozhodol implementovať pomocou technológie .NET v programovacom jazyku C# s databázovým serverom Microsoft SQL. Pre prístup k dátam som používal technológiu .NET Frameworku nazvanú ADO.NET, ktorá umožňuje aplikáciám pripojiť sa k zdroju dát a manipulovať s ním. Preto som si vybral práve tieto technológie a tento jazyk, lebo sú to pre mňa nové technológie s ktorými som sa chcel zoznámiť pre ďalšie skúsenosti s programovaním. Ďalším veľkým dôvodom pre túto voľbu bola aj kvalitná dokumentácia od firmy Microsoft a začiatočná komunita používateľov pre tieto technológie. Ako vývojové prostredie som si vybral tiež produkt Microsoftu a to Visual Studio 2005.

6.1.1 .NET Framework

.NET Framework je prostredie od firmy Microsoft, potrebné pre beh .NET aplikácií na osobných počítačoch s operačným systémom Windows a to od verzie Windows 98. Obsahuje tak ako spúšťačie rozhranie aj všetky potrebné knižnice.

.NET je možné chápať ako zastrešujúci názov pre súbor technológií v softvérových produktoch, ktoré tvoria celú platformu, ktorá je dostupná nielen pre Web, Windows a Pocket PC. .NET nie je viazaný len na Windows, umožňuje spúšťať aj všetky .NET aplikácie na unixových platformách s použitím potrebných komponentov. Na všetkých platformách .NET zdieľa rovnaké základné princípy, čo robí prechod medzi nimi jednoduchším. Rôzne verzie .NET Frameworku sú súčasťou novších verzií operačného systému Windows.

Aktuálna verzia 3.5 je pre majiteľov operačného systému Windows k dispozícii zdarma ako samostatná komponenta, ktorá sa do systému doinštaluje. Platforma .NET nepredpisuje použitie žiadneho programovacieho jazyka. Bez ohľadu na to, v čom bola aplikácia pôvodne napísaná, sa vždy preloží do medzijazyka Common Intermediate Language. Najpoužívanější programovacie jazyky pre vývoj .NET aplikácií sú C#, Visual Basic .NET a Delphi.

6.1.2 Jazyk C#

C# je objektovo - orientovaný programovací jazyk vyvinutý spoločnosťou Microsoft ako časť ich iniciatívy .NET. Microsoft si za základ pre nový jazyk C# zobral C++ a jazyk Java. C# bolo navrhované s úmyslom vyvážiť silu jazyka C++ a tú spojiť s možnosťou rýchleho programovania, ktoré ponúkali jazyky ako napríklad Visual Basic, Delphi.

Jednou z najznámejších vylepšení oproti jazyka C++ je automatický garbage collector. Stará sa o správu pamäti tak, že uvoľňuje už nepoužívanú pamäť a udržuje tak voľné systémové prostriedky dostupné pre ďalšie aplikácie. C# ako programovací jazyk v niektorých ohľadoch priamo odráža vlastnosti vrstvy CLI (Common Language Infrastructure), ktorá leží pod ním. Tento jazyk bol priamo navrhnutý tak, aby umožňoval využitie všetkých vlastností, ktoré poskytuje CLI, na rozdiel od jazykov, ktoré majú vlastnú syntaxu a využívajú len podmnožinu vlastností CLI (ako napríklad Visual Basic). Väčšina typov zavedených v jazyku C# priamo korešponduje s hodnotovými typmi implementovanými v CLI framework, špecifikácia jazyka C# však neurčuje podmienky, ktorými sa má generovať kód z kompilátora. To znamená, že kompilátor jazyka C# nemusí mať za cieľovú podpornú platformu priamo CLI. C# je jazyk case - sensitive tak isto ako všetky moderné objektovo orientované jazyky. V tomto jazyku je aj realizovaný väčší počet základných knižníc .NET framework [14].

6.1.3 MS SQL Server

MS SQL Server je veľký a spoľahlivý databázový systém, ktorý tvorí databázovú vrstvu pre aplikácie podnikových informačných systémov strategického významu. Jeho nasadenie ako databázy pre interaktívne internetové aplikácie je tak isto bežné.

Databázová platforma MS SQL je dostupná len pre operačné systémy Microsoft Windows. Na verziách bez technológií NT (Windows 95 / 98 / Millenium Edition) je možné prevádzkovať MS SQL Server Personal Edition, alebo jadro MSDE. Pre plné využitie je potrebný operačný systém Windows 2000, XP alebo Vista. Každá verzia databázového serveru je dodávaná s užitočnými aplikáciami s grafickým používateľským rozhraním, uľahčujúc vývojárovi jeho prácu. Je navrhnutý tak, aby podnikom pomohol úlohy vyriešiť.

Pre správu a analýzu dát poskytuje zvýšené zabezpečenie, dostupnosť podnikových dát a analytických aplikácií. Súčasne uľahčuje ich vytváranie, nasadenie a správu. Umožňuje zdieľať dáta medzi viacerými platformami, aplikáciami a zariadeniami a uľahčuje tak prepojenie interných a externých systémov. Je v ňom k dispozícii aj rozhranie pre import a export dát, čo sa hodí pri prechode z iného alebo na iný systém. Aplikácia pre správu databázy umožňuje správcovi všetky úlohy administrácie serveru – správu rolí, používateľov a ich prístupových práv, pravidiel pre zálohovanie a údržbu.

6.1.4 Microsoft Visual Studio 2005

Microsoft Visual Studio je integrované vývojové prostredie pre vývoj aplikácií, ako formulárových tak aj webových služieb a stránok. Podporuje mnoho programovacích jazykov, medzi nimi aj C#, ktorý bol používaný aj pri vývoji systému pre riadenie rizík.

Poskytuje pokročilé vývojové nástroje, ladiace funkcie a funkcie pre prácu s databázami pre rýchlu tvorbu špičkových aplikácií. Obsahuje vizuálne návrháre pre vývoj s využitím .NET Framework 3.5 a zdokonalené nástroje pre lepšiu spoluprácu vo vývojárskych tímoch. Sú to predovšetkým nástroje, ktoré pomáhajú zapojiť do vývojového procesu aj databázových

odborníkov a návrhárov grafiky. Poskytuje aj stavebné bloky pre riešenia častých programátorských úloh. Užitočnou súčasťou je nástroj pre návrh formulárov pre aplikácie s grafickým používateľským rozhraním. Visual Studio znižuje potrebu každodenného písania programového kódu, skraca čas vývoja a dovoľuje vývojárom sústrediť sa na riešenia vlastného problému [15].

6.1.5 ADO .NET

V dnešnej dobe je veľké množstvo aplikácií, či už desktopových alebo webových, napojených na nejakú databázu, z ktorej získavajú, zobrazujú a modifikujú dáta. .NET Framework obsahuje vlastnú technológiu pre prístup k dátam nazvanú ADO.NET, ktorá umožňuje aplikáciám pripojiť sa k zdroju dát a manipulovať s ním.

ADO.NET oddeľuje činitele prístupu k dátam a činitele manipulácie s dátami do oddelených komponentov, ktoré môžu byť použité samostatne alebo spojit. Technológia ADO.NET zahŕňa poskytovateľa dát pre pripojenie k databáze, vykonávanie príkazov a získavanie výsledkov. Tieto výsledky sú buď spracované priamo alebo sú umiestnené v objekte DataSet, aby boli odkryté kombinovane s dátami z viacerých zdrojov alebo vzdialene medzi dvoma vrstvami. Objekt DataSet je možné použiť aj nezávisle od poskytovateľa dát .NET Frameworku, aby spracovával dáta lokálne pre aplikáciu alebo zdrojovo z XML. ADO.NET používa viacvrstvovú architektúru založenú na poskytovateľoch dát. Je to sada tried, ktoré sú určené na prístup ku konkrétnej databáze, vykonávanie SQL príkazov a získavanie dát.

Do týchto tried patrí [16]:

- **connection** - zriaďuje spojenie so zdrojom dát,
- **command** - vykonáva SQL príkazy a uložené procedúry,
- **dataReader** - poskytuje rýchly prístup (iba na čítanie smerom vpred) k dátam získaným z dotazu,
- **dataAdapter** - napĺňa sadu dát (DataSet) a premieta zmeny do zdroja dát.

ADO.NET neobsahuje obecné objekty určené na manipuláciu so všetkými druhmi databáz, ale poskytovateľa dát, ktorý je navrhnutý konkrétne pre rôzne druhy zdrojov dát, medzi ktoré patrí SQL Server, OLE DB, Oracle a ODBC [16].

6.2 Implementácia databázy

Databázová štruktúra systému je implementovaná v databázovom serveru MS SQL (viď. Kapitola 6.1.3 – MS SQL Server). Pre správny beh systému musíme mať nainštalovaný tento server aj s potrebnou konfiguráciou.

6.2.1 Vytvorenie databázových tabuliek

Vzťahy, referencie a atribúty sú implementované podľa ER diagramu (viď. Kapitola 5.4 – Návrh databázy), ktorý sa skladá z 13 entít. Každá entita ER diagramu reprezentuje samostatnú tabuľku v databázovej štruktúre a jednotlivé vzťahy sú riešené pomocou cudzích kľúčov (*FOREIGN KEY*).

Pre prácu z databázovými tabuľkami existujú dva typy kľúčov: primárny a cudzí kľúč. Každá tabuľka obsahuje jeden primárny kľúč a ten je v prípade vzťahu s inou tabuľkou spojený ako cudzí kľúč. Primárny kľúč musí mať veľmi dôležitú vlastnosť, jeho hodnota musí byť v celej tabuľke jedinečná. Keď nastane chyba a primárny kľúč nebude jedinečný nastáva kolízia dát. Primárny kľúč značne pomáha pri práci s databázou pri väčšom objeme dát a označuje sa ako „*PRIMARY KEY*“. Hlavným parametrom je „*AUTO INCREMENT*“, ktorý zabezpečuje to, že pri vložení hodnoty 0 sa

vygeneruje číslo o 1 vyššie, než je najvyššie číslo v tomto stĺpci, čím dosiahnem, že každý riadok bude mať unikátne číslo. Vytvorenie databázy môže prebiehať dvojakým spôsobom:

1. Vytvorenie pomocou skripta.
2. Vytvorenie pomocou príkazov.
3. Vytvorenie graficky, priamo v integrovanom vývojovom prostredí.

Prvý spôsob vytvorenia sa prejavuje v tom, že je napísaný malý skript (programový kód), ktorý je spustený na databázovom servere. Tento skript obsahuje popis dátovej štruktúry, definíciu dátových polí, obmedzenia a pridávané údaje. Do skripta sa píše tie príkazy, ktoré môžu byť písané aj do príkazového riadku databázového serveru. Toto definuje druhý bod vymenovania, kde príkazy používané v skripte sú zadané jednotlivo.

Pri týchto možnostiach sa používajú príkazy jazyka SQL. Napríklad pre vytvorenie tabuľky, ktorá by zodpovedala entite ER diagramu sa používa príkaz „*CREATE TABLE*“, za ktorým musí nasledovať názov tabuľky a v zátvorke jednotlivé stĺpce tabuľky oddelené čiarkou, pričom prvý je názov stĺpca a za ním je jeho typ. Typy hodnôt môžu byť napríklad: *INT*, *VARCHAR*, *DECIMAL* atď. Pri vytvorení stĺpca je možné určiť, či daný stĺpec môže obsahovať prázdne hodnoty alebo nie. Keď chceme, aby záznam mal v každom prípade nejakú hodnotu, priradíme do popisu „*NOT NULL*“. Je možné pri tvorbe novej tabuľky špecifikovať aj implicitnú hodnotu s modifikátorom „*DEFAULT*“.

Tretí spôsob vytvorenia databázovej štruktúry je podľa môjho názoru jednoduchší a efektívnejší. Priamo v grafickom rozhraní SQL Serveru, alebo v integrovanom vývojovom prostredí sú nakreslené diagramy, ktoré by potom zodpovedali entitám ER diagramu. Používateľ má pred sebou prehľadne znázornené všetky tabuľky, ktoré je potom možné vyplniť s atribútmi. Atribúty primárneho kľúča každej tabuľky je treba označiť obvykle zvlášť a treba si nastaviť aj jeho inkrementáciu. Vždy je možné vytvoriť s jednoduchým spojovaním týchto tabuliek, kde spájať je možné primárny kľúč jednej tabuľky s cudzím kľúčom druhej tabuľky. Vo vývojovom prostredí je možné tieto tabuľky priamo naplniť si údajmi, s jednoduchým vyplňovaním riadkov tabuliek, podľa zadaných stĺpcov. Z takto vytvorenej databázy je možné vygenerovať si skript obsahujúci štruktúru a údaje databázy. Tento skript je potom identický skriptu, popísaný v predchádzajúcich odstavcoch.

6.2.2 Prístup k dátovej vrstve

Pristupovať k dátam pomocou menného priestoru ADO.NET (viď. Kapitola 6.1.5), ktorý som používal pri implementácii systému, je možné dvojakým spôsobom. Je možné používať buď pripojenú alebo odpojenú koncepčnú charakteristiku. Oba tieto charakteristiky boli používané pri vývoji systému pre riadenie rizík v IT.

Keď bola používaná pripojená vrstva, báza kódov sa explicitne pripojovala k podkladovému úložisku dát a odpojovala sa od nej. Táto komunikácia prebiehala typicky pomocou objektov pripojenia, príkazov a čítačov dát. Čítači dát poskytujú spôsob, pomocou ktorých sa dajú získať záznamy z úložiska dát prístupom, ktorý môže postupovať len smerom dopredu a ktorý je určený k čítaniu. S pripojenou vrstvou som pracoval v prípadoch, kedy bolo treba do databázy uložiť alebo zmeniť údaje.

V prípadoch, kedy bolo treba z databázy len načítať údaje som pracoval s odpojenou vrstvou. Odpojená vrstva dovoľuje získať množinu dátových tabuliek (objekty *DataTable* obsadené vnútri nejakej sady dát, *DataSet*), ktorá funguje ako kópia externých dát u klienta. Keď pomocou asociovaného objektu dátového adaptéra sa získava sada dát (*DataSet*), pripojenie sa automaticky otvorí a uzavrie. Takýto prístup umožňuje rýchlo uvoľňovať pripojenie ďalším volajúcim. Akonáhle

je získaná sada dát, je možné s jej obsahom manipulovať, bez toho aby boli využité sieťové prepojenia s databázovým serverom. Keď je treba odoslať prevedené zmeny späť do úložiska dát, opäť sa použije dátový adaptér, aktualizuje sa zdroj dát a spojenie sa okamžite uzavrie.

Každý databázový poskytovateľ je uložený vo vlastnom mennom priestore, ktorý je podprieštorm *System.Data*. V mojom prípade, keďže som pracoval s Microsoft SQL Serverom, som používal triedu uloženú v mennom priestore *System.Data.SqlClient*, ktorá začínala predponou *Sql*.

K pripojeniu databázy MS Sql Server som používal triedu *SqlConnection*. Ako argument konštruktoru tejto triedy je potrebné zadať takzvaný "prepojovací reťazec", čo je reťazec definujúci atribúty prepojenia. Pre pripojenie k SQL Serveru, ktorý je sprostredkovaný triedou *System.Data.SqlClient.SqlConnection*, je nevyhnutné poznať tieto údaje. Sú to údaje obsahujúce dvojicu názov = hodnota. Prepojovací reťazec môžeme triede priradiť v konštruktoru tejto triedy, či pomocou vlastnosti *ConnectionString* (z ktorej ju potom môžeme späťne získať). Prepojovací reťazec môže obsahovať radu nastavení pripojenia, napríklad názov servera, meno a heslo používateľa databázy a veľkosť balenia prenášajúcu dáta či dobu, koľko má aplikácia čakať na odozvu SQL Serveru. Nasledujúce vymenovanie popisuje povinné údaje prepojovacieho reťazca:

- **server** – „Data Source“, predstavuje cestu či adresu databázy,
- **database** – „InitialCatalog“, predstavuje názov databázy,
- **uid** – „UserID“, predstavuje meno používateľa databázy,
- **pwd** – „Password“, predstavuje heslo používateľa databázy.

Výsledný prepojovací reťazec by mohol vyzeráť nasledujúcim spôsobom:

```
string prepRetazec = " server = localhost; database = Rizika; uid = user;
pwd = password ";
```

Tento prepojovací reťazec potom musíme priradiť k inštancii triedy *SqlConnection* buď priamo v konštruktoze, alebo pomocou vlastnosti *ConnectionString*. Je to možné len v prípade, že *SqlConnection* nie je k databáze pripojený.

```
SqlConnection sqlConn = new SqlConnection(prepareRetazec);
SqlConnection sqlConn = new SqlConnection ();
sqlConn.ConnectionString = prepRetazec;
```

V prípade ak som používal databázu v odpojenej podobe, bolo treba urobiť kópiu externých dát a načítať to do množiny dátových tabuliek (*DataSet*). Načítanie prebiehalo vykonaním nejakého SQL príkazu, ktorého výsledok bol uložený do *DataTable*. Pri načítaní si najprv treba vytvoriť pripojenie, čo je predstavované inštanciou triedy **Connection*, ktorú je možné následne otvoriť metódou *Open* tejto inštancie. Potom je treba vytvoriť dátový adaptér a jeho konštruktoru predať reťazec predstavujúci dotaz SQL a už vytvorené pripojenie k dátovému zdroju. Dátová množina, ktorá je predstavovaná inštanciou triedy *System.Data.DataSet*, je určená k uchovávaní dát získaných dátovým adaptérom. K naplneniu používam metódu **DataAdapterFill*, ktorá očakáva ako svoj argument práve tento *DataSet* a názov novej tabuľky.

```
string prikaz = "SELECT * FROM Rizika";
SqlDataAdapter adapter = new SqlDataAdapter(prikaz, pripojenie);
DataSet deset = new DataSet();
Adapter.Fill(dSet, "Rizika");
```

Pre prácu s celou databázou ako odpojenou vrstvou, som vytvoril pre všetky tabuľky nové objekty *DataTable* v novom *DataSete*. Vytvorenie tabuliek sú implementované ako nová trieda systému, ktorá je volaná pri spustení aplikácie. Vtedy je vytvorený nový *DataSet* s potrebnými dátovými tabuľkami a naplnený s údajmi z pôvodnej databázy. Takto je vytvorených 13 dátových tabuliek s údajmi, ktoré sú identické s tabuľkami uloženými na databázovom serveri.

Ako z dátovej sady získať dáta, ako ich aktualizovať a naplniť je podrobne popísané v literatúre [16]. Sú tam popísané aj práce s pripojenou vrstvou, s ktorou aplikácia pracuje oveľa menej krát, aby zbytočne nezaťažovala sieť, cez ktorú bude pripojená k SQL serveru.

ADO.NET umožňuje aj niektoré komponenty z grafického používateľského rozhrania viazať na dáta z dátovej sady. Túto možnosť zobrazenia dát som používal hlavne tam, kde bolo treba dáta zobrazovať v tabuľke alebo v roletovom menu. Keď je niektorý z prvkov viazaný na stĺpec v dátovej sade, hodnota stĺpca sa v danom prvku zobrazuje prostredníctvom vlastnosti *Text* v skupine *DataBindings*, ktorá nastavuje text prvku. Vlastné viazanie a nastavenie je možné previesť buď vo vývojovom prostredí, alebo priamo naprogramovať do kódu. Viazanie komponentov som zadával pomocou vlastnosti *DataSource* a zobrazenie pomocou *DataBindings*.

6.3 Hlavné funkcie systému

V tejto časti detailne opíšem implementované funkcie systému pre riadenie rizík v IT, ktoré boli z celá realizované podľa analýzy a návrhu (viď. Kapitola 5). Budú vysvetlené aj funkcie, ktoré boli implementované ako na aplikačnej, tak na používateľskej vrstve.

Pretože k systému pristupujú používatelia s rôznymi právami, budú u jednotlivých funkcií vysvetlené aj ich role. Tieto funkcie som implementoval v jazyku C# (viď. Kapitola 6.1.2). Pri implementácii boli využívané princípy objektovo orientovaného programovania a dodržiované aj moderné prístupy k tvorbe desktopových aplikácií. Tiež boli dôsledne dodržiované rozdelenia jednotlivých vrstiev podľa návrhu architektúry (viď. Kapitola 5.3).

6.3.1 Prihlásenie a odhlásenie zo systému

Po spustení aplikácie je používateľ vždy vyzvaný na bezpečné prihlásenie do systému (Obrázok 6.1). Musí vyplniť svoje prihlasovacie meno a heslo, ktoré mu prideliť administrátor pri registrovaní do systému. Registrácia bude popísaná v ďalšej kapitole.



Obrázok 6.1: Prihlasovanie do systému

Po úspešnom vyplnení údajov je formulár poslaný na spracovanie. Spracovanie prihlásenia je implementované v triede *Login.cs*. Systém sa pokúsi nájsť prihlasovacie meno v databáze používateľov, ktorí ich jednoznačne identifikuje. Z tejto skutočnosti vyplýva, že login – prihlasovacie meno každého používateľa musí byť unikátne. V prípade, že systém prihlasovacie meno nájde, kontroluje jemu priradené heslo s heslom zadanom do formulára. Samozrejme heslo zadané do formulára je zakódované pomocou MD5 kódovania a tak je porovnávané s údajom v databáze. V databáze sú heslá zakódované pomocou hash - e MD5 z bezpečnostných dôvodov. V prípade, že systém zadané prihlasovacie meno nenájde v databáze, alebo zadané heslo nie je ekvivalentné s heslom priradeným k loginu, zobrazí sa chybové hlásenie.

Keď používateľ pošle k systému na spracovanie prázdny formulár, bude vyvolaná výnimka, ktorá je ošetrená výpisom chybového hlásenia. Po zhodách používateľského mena a hesla je používateľ spustený do systému a môže s ním ďalej pracovať. Pri činnosti prihlasovania eviduje systém právo prihláseného používateľa a podľa tejto hodnoty mu umožňuje systémové funkcie ekvivalentné jeho právam.

Odhlásenie je používateľovi umožnené všade v systéme, ktoré môže urobiť kliknutím na adekvátnu ikonu, alebo si môže vybrať z hornej menu aplikácie. Po odhlásení systém vynuluje svoju štruktúru o prihlásenom používateľovi a dostane sa do stavu čakania na nové prihlásenie.

6.3.2 Pridávanie a správa používateľov

Pridávanie a správu používateľov má na starosti ten používateľ systému, ktorý disponuje právom administrátora. Ten má ako jediný prístup do tejto sekcie a je jediný v systéme, ktorý môže zmeniť údaje všetkých používateľov. Administrátori systému sú obvykle vedúce osoby firmy, alebo nimi vymenované kompetentné osoby práve na túto činnosť.

Táto sekcia systému, ktorá je zobrazená na obrázku 6.2 obsahuje pridávanie, zmenu a zmazanie používateľov.

Správa pracovníku

	id_pracovníka	meno	priezvisko	login	funkcia	email
	2	John	Lock	lock	projektový manažér	dvrn@sdsnj.sk
	4	Jackk	Shephard	jack	projektový manažér	dvddbv
▶	7	Jan	Novák	xnovak	administrátor	novak@firmmail.cz
	8	Petr	Nagy	xnagy	projektový manažér	petr.nagy@sezne
	9	Ludmila	Voková	xvokova	programátorka	donald@azet.sk
	10	Jana	Mala	xmala	programátorka	mala@seznam.cz

Zmena Pridat

Pridat pracovníka:

Meno: Příjmení:

Login: Heslo:

Právo: Email:

Funkce:

Nadřazení:

☐ Má nadřazeného?

Přidat Ukončit

Obrázok 6.2: Správa používateľov systému

V hornej časti okna je tabuľka všetkých používateľov, ktorí sú v systéme zaregistrovaní. V prípade, keď administrátor chce zmeniť niektoré údaje o pracovníkovi, vyberie si toho pracovníka

z tabuľky. V tomto prípade príde k udalosti výberu riadku tabuľky a systém automaticky vypíše do formulára všetky údaje tohto pracovníka. Je to implementované pomocou ovládača *CellClick*, ktorý zisťuje kliknutie na riadok tabuľky a vyvoláva ošetrujúcu funkciu. Po tomto kliknutí už budú administrátorovi k dispozícii údaje, ktoré môže meniť bez obmedzenia. Takým istým spôsobom sa prevedie aj zmazanie pracovníkov.

Pridanie nového pracovníka je možné vyplnením formulára, ktorý obsahuje údaje:

- povinné,
- nepovinné.

Jediným nepovinným údajom je nadriadený pracovník pridávaného pracovníka. V prípade zadania tejto hodnoty sa musí vybrať z roletového menu prihlasovacie meno nadriadeného pracovníka. Keď administrátor chce, aby zadanie bolo úspešné, musí vyplniť všetky povinné položky vo formulári. V prípade, že sa zabudne vyplniť niektorý s povinných údajov, objaví sa chybové hlásenie. Každá položka vo formulári má nastavené určité obmedzenia, ako je napr. počet zadávaných znakov.

Po odoslaní údajov musí byť testované, či boli zadané všetky povinné údaje a v prípade chyby vypísať hlásenie tejto skutočnosti. Keď boli zadané všetky povinné údaje systém začína kontrolu významu údajov. Pod týmto pojmom sa rozumie kontrola prihlasovacieho mena používateľa. Je potrebné kontrolovať, či v systéme neexistuje používateľ pod takýmto prihlasovacím menom. Táto kontrola sa prevedie s pokusom vybrať si z databázy používateľa zo zadaním prihlasovacím menom. Keď výber bude mať platnú hodnotu, login je už v systéme registrovaný.

Keď už všetky kontroly prebehli úspešne, používateľ je pridaný do databázy s uvedenými údajmi. Aktualizuje sa aj tabuľka nad formulármi, v ktorom už používateľ bude znázornený.

6.3.3 Výber a otvorenie projektov

Po úspešnom prihlásení do systému používateľ bude mať k dispozícii formulár zobrazený na obrázku 6.3 pre výber projektu.

The screenshot shows a web-based application window titled "Projekty". It features a sidebar with a folder icon and a globe. The main content area is divided into several sections. The top section, "Výběr projektu:", includes a user login status "Přihlášení uživatel: Jan Novak administrátor" and a project selection area "Projekty:" with a checked checkbox "jenom projekty na které jsem přiřazen" and a dropdown menu "Vytvoření systému pro řízení rizik". There are two links: "Vytvořit projekt" and "Správa uživatelů". Below this is the "Informace o projektu:" section, which contains input fields for "Název projektu:" (filled with "Vytvoření systému pro řízení rizik v IT"), "Popis:" (filled with "Vytvoření kompletního systému pro řízení rizik v rámci vývoji systémů Informačních technologií..."), "Vedoucí projektu:" (filled with "Petr Nagy"), and "Začátek:" (filled with "8.5.2009"). At the bottom left, there is a logo with the Greek letter pi and the text "Výpočet EMV". At the bottom right, there are two buttons: "OK" and "Ukončit".

Obrázok 6.3: Výber projektu

Používateľovi sú implicitne ponúkané tie projekty, na ktoré je priradený ako účastník. So zmenou zaškrtnutia políčka je možné zmeniť údaje na zobrazenie všetkých projektov. Používateľovi tento maličký filter umožňuje mať prehľad len o tých projektoch, kde je zainteresovaný. Z pohľadu implementácie bolo treba skontrolovať tieto komponenty a prípadné zmeny prepisovať na vybrané hodnoty. Na zmenu hodnoty zaškrtnutia políčka slúži spracovávacia udalosť *CheckedChanged*. V prípade volania tejto funkcie je obsah roletového menu obsahujúci projekty zmenený na všetky projekty uložené v systéme. SQL dotaz je zmenený z implicitného výberu s obmedzením na výber bez obmedzenia.

Z tejto sekcie systému je možné pre oprávnených používateľov dostať sa aj do správy pracovníkov (viď. Kapitola 6.3.2), alebo založiť si nový projekt v prípade projektového manažéra. Ešte však pred samotným založením nejakého projektu, systém umožňuje aj rozhodovanie medzi projektmi podľa očakávanej peňažnej hodnoty. Túto funkciu systému popisuje nasledujúca kapitola.

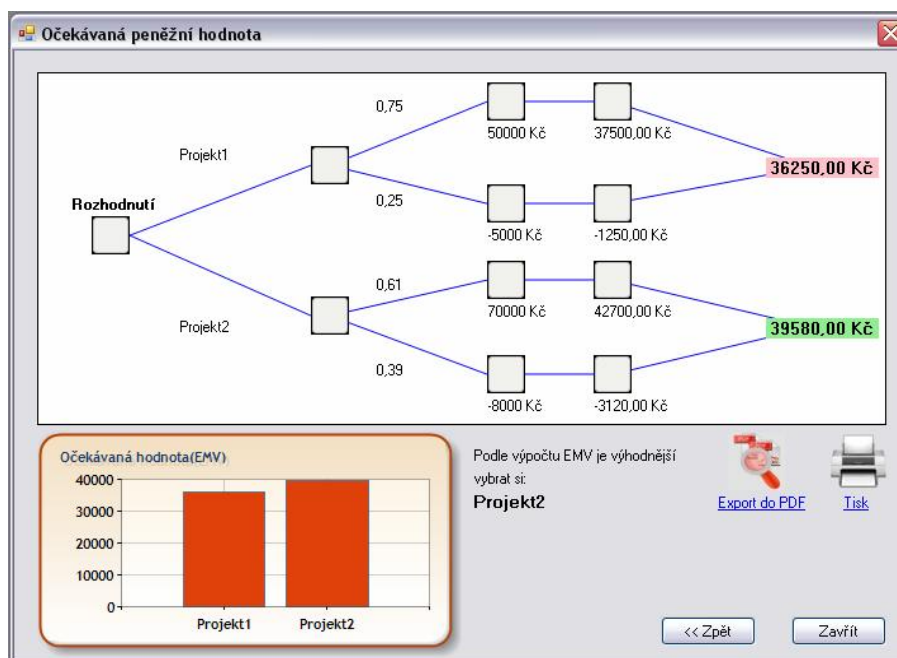
6.3.4 Výpočet očakávanej peňažnej hodnoty

Táto funkcia systému pomáha vedúcim pracovníkom vybrať si projekt s najväčšou peňažnou hodnotou. Tento výpočet nie je viazaný len na projekty, ale umožňuje rozhodovanie aj v iných fázach projektu. Môže to byť napríklad rozhodovanie medzi pesimistickým plánom a optimistickým plánom pred pokračovaním fázy.

Prvý krok výpočtu je formulár pre vstupné údaje. Systém umožňuje rozhodovanie medzi dvoma udalosťami, preto je potrebné pre oba zadať nasledujúce hodnoty:

- očakávaná peňažná hodnota pri úspechu,
- očakávaná peňažná strata pri neúspechu,
- pravdepodobnosť vzniku úspechu.

Pre správny priebeh výpočtu je potrebný aj údaj o pravdepodobnosti vzniku neúspechu. Keďže táto hodnota je jednoducho vyvoditeľná z predchádzajúcich údajov, systém zbytočne používateľa nezaťažuje s jeho vyplnením. Po správnom zadaní týchto údajov prebehne výpočet, ktorý je vidieť na obrázku 6.4.



Obrázok 6.4: Výpočet očakávanej peňažnej hodnoty

Pravdepodobnosť vzniku možných udalostí je potrebné odhadnúť pomocou expertného posudzovania. Toto posudzovanie v praxi odhadnú vedúci pracovníci, alebo ľudia, ktorí majú s riešením podobných projektov nejaké skúsenosti a dokážu teda odhadnúť, ako veľké sú šance na úspech.

Pre výpočet očakávanej peňažnej hodnoty oboch projektov stačí u každého z uvedených výsledkov vynásobiť potenciálnu peňažnú hodnotu s pravdepodobnosťou jej vzniku a tieto odlišné hodnoty k jednému projektu spočítať. Keďže očakávaná peňažná hodnota predstavuje odhad výslednej peňažnej hodnoty určitého rozhodnutia, je žiadajúce dosiahnuť pozitívny výpočet. Čím vyššia je očakávaná hodnota, tým je to lepšie. Do výpočtu zahrňujeme všetky možné výsledky spolu ich pravdepodobnosťami a preto pri rozhodovaní oslabíme prirodzenú tendenciu niektorých ľudí presadiť si príliš agresívnu alebo naopak príliš konzervatívnu stratégiu riadenia rizík.

Ako je vidieť na obrázku 6.4 okrem rozhodovacieho stromu, výsledok obsahuje aj graf výstupu. Tento graf obsahuje výslednú peňažnú hodnotu oboch projektov a takto umožňuje jednoduché vyjadrenie výsledku rozhodnutia. Výsledky je možné vyexportovať aj do PDF súboru pre neskoršie spracovanie, alebo priamo vytlačiť cez tlačiareň.

6.3.5 Vytvorenie nového projektu

Vytvoriť projekt je možné len pracovníkom s právom projektového manažéra alebo s vyšším právom. Môžu to urobiť hneď po prihlásení do systému, alebo aj pri práci s iným projektom. Samozrejme rozpracovaný projekt a všetky jeho úlohy budú ukončené.

Samotné pridávanie projektu zahrňuje v sebe vyplnenie formulára s potrebnými údajmi o projekte a priradenie pracovníkov na projekt. Formulár, ktorý je potrebné vyplniť obsahuje povinné údaje ktoré sú:

- názov,
- popis,
- stav,
- začiatok.

Okrem povinných údajov môže používateľov projekt spresniť aj s nepovinnými údajmi:

- koniec projektu,
- náklady na projekt.

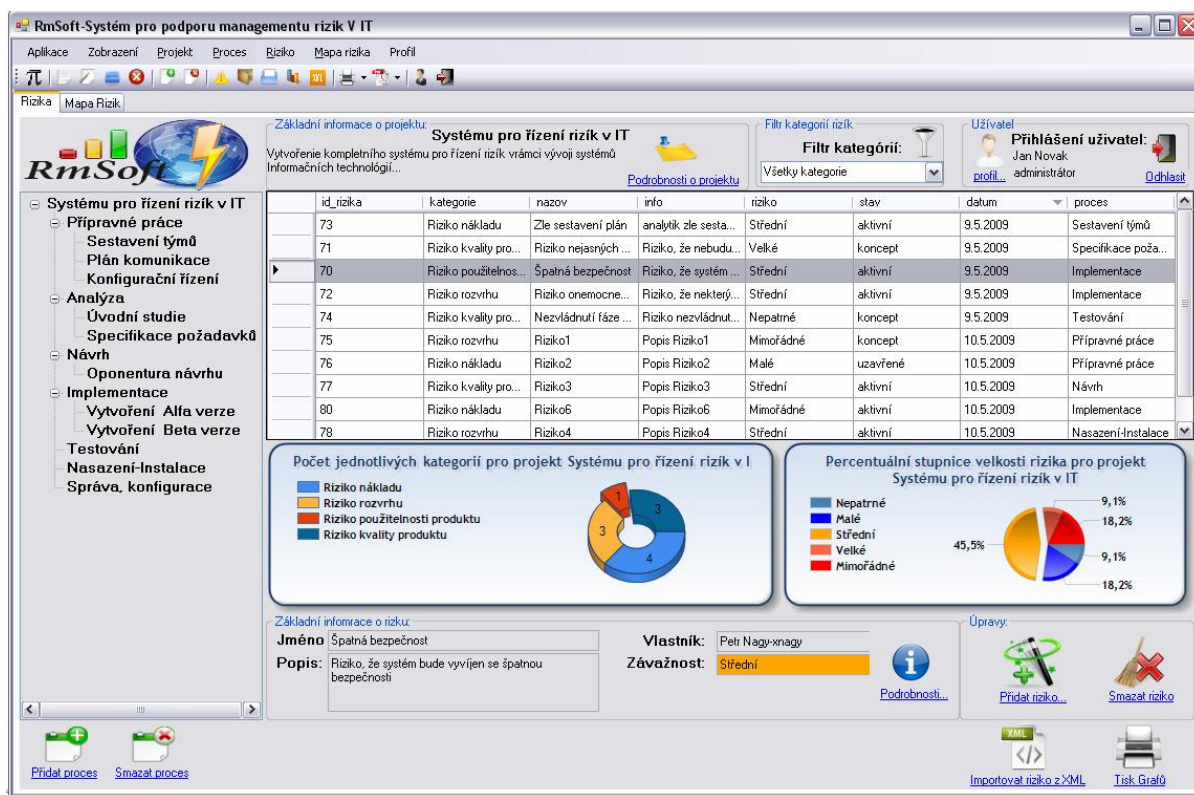
Po úspešnom vyplnení budú tieto údaje uložené v databáze ako atribúty tabuľky projekt. Systém kontroluje správnosť zadaných hodnôt rôznymi spôsobmi. Napríklad, aby používateľ nemohol zadať na začiatok projektu nezmyselnú hodnotu dátumu, je použitá komponenta *dateTimePicker*. Táto komponenta umožňuje zadať len hodnotu platného dátumu, v takom formáte, ktorý je jemu pridelený. Formát, ktorý používam v celom systéme je takzvaný „krátky formát dátumu“ (*short date format*), kde sú zobrazené len dni, mesiace a roky bez časových hodnôt.

Priradiť pracovníka k projektu je možné výberom z tabuľky všetkých pracovníkov. Tabuľka, v ktorej sú zobrazené údaje o pracovníkoch je odvodená z komponentu *dataGridView*. Táto komponenta umožňuje jednoducho a prehľadne pracovať s údajmi zobrazenými vo forme tabuľky.

Pracovník, ktorý projekt vytvára bude automaticky priradený k tomuto projektu. Je bezpečne kontrolovaná aj nemožnosť zadania projektu bez priradených pracovníkov. Zadať projekt bez projektového manažéra je ale možné, pritom neskoršie úpravy tímu tohto projektu musí už previesť administrátor systému. Po úspešnom dokončení bude projekt automaticky vybraný zo zoznamu projektov a umožnený na riadenie rizík.

6.3.6 Rozhranie projektu a jeho rizík

Práca s projektom a rizikami zahŕňa v sebe hlavné rozhranie aplikácie, s ktorým bude mať celý čas používateľ kontakt pri riadení rizík. Hlavné rozhranie aplikácie je vidieť na obrázku 6.5.



Obrázok 6.5: Hlavné rozhranie aplikácie

Toto rozhranie, ktoré je vidieť na obrázku 6.5 definuje prácu s konkrétnym projektom, kde je definované aj riadenie rizík. Grafické rozhranie je implementované podľa návrhu a je rozdelené na viac častí. Hlavnou časťou je stromová štruktúra obsadená na ľavej strane systému. Tento strom ukazuje jednotlivé procesy projektu, ktoré môže obsahovať aj ďalšie zanorené procesy. Pre tieto procesy sú definované riziká, ktoré sú znázornené v tabuľke, obsadené v strede okna. Keď používateľ klikne na procese stromu, v tabuľke sa mu zobrazia len tie riziká, ktoré patria k danému procesu. V prípade kliknutia na meno projektu, čo je v štruktúre obsadené na vrchole stromu, systém zobrazí všetky riziká týkajúce sa celého projektu. Nad tabuľkou je výpis informácií o projekte, alebo o procese podľa toho, ktorá úroveň stromu je vybraná. Riziká zobrazené v tabuľke je možné zoradovať podľa všetkých stĺpcov, stačí len kliknúť si na daný stĺpec. Používateľovi je k dispozícii aj filter kategórií. Tento filter umožňuje zobrazovať len tú kategóriu rizika pre daný proces, alebo na celý projekt, ktorá je v danej chvíli pre riadenie dôležitá.

Pod tabuľkou rizík sa nachádzajú dva grafy pre prehľadnejšie zobrazenie projektu definovaných rizík. Prvý graf je typu prstencového a zobrazuje počet rizík jednotlivých kategórií. Druhý graf zase zobrazuje percentuálne rozloženie závažností týchto rizík. V prípade ak používateľ klikne na jednom riziku, z grafov sa vyčlení tá časť, v ktorej je dané riziko obsadené. Okrem prekreslenia grafov sa zobrazujú aj základné informácie týkajúce sa rizika a s týmto rizikom sa môže ďalej pracovať. Tieto štatistiky umožňujú projektovému manažérovi mať všeobecný prehľad o všetkých rizikách identifikovaných v projekte. Tieto informácie je možné vytlačiť cez tlačiareň spolu so základnými informáciami ako je meno projektu, popis a vedúci projektu.

Z tejto sekcie je používateľovi umožnená aj celková správa všetkých komponentov systému ako je pridávanie procesov k projektu, správa informácií o projekte a všetky práce s rizikami. Patria sem aj najzákladnejšie funkcie informačných systémov ako odhlásenie, profil používateľa s možnosťou zmeny údajov a samozrejme ukončenie práce s aktuálnym projektom. Funkčnosť týchto častí systému budú popísané v ďalších kapitolách tejto práce.

6.3.7 Správa projektu

Spravovať projekt môže okrem administrátora aj projektový manažér, ktorý daný projekt vytvoril. Táto správa v sebe obsahuje dve stránky spravovania.

Údaje o projekte

Údaje | Pracovníci

Údaje o projekte

Povinné údaje:

Název: Systému pro řízení rizik v IT

Popis: Vytvoření kompletního systému pro řízení rizik v rámci vývoje systémů informačních technologií...

Stav: Aktivní

Začátek: 8. 5. 2009

Nepovinné údaje:

Konec: 30. 9. 2009 ☒ Zadat konec

Náklady: 150000.00

Změnit Zrušit

Obrázok 6.6 a) Správa projektu - informácie

Údaje o projekte

Údaje | Pracovníci

Pracovníci

Pracovníci v systému

login	meno	prijmeni	funkce
novak	Jan	Novak	administrátor
lock	John	Lock	projektový manažér
jack	Jack	Shephard	projektový manažér
xnovak	Jan	Novák	administrátor
xnagy	Petr	Nagy	projektový manažér
xvokova	Ludmila	Voková	programátorka
xmala	Jana	Mala	programátorka

Přirazení pracovníci k projektu

login	meno	prijmeni	funkce
novak	Jan	Novak	administrátor
xvokova	Ludmila	Voková	programátorka
xnagy	Petr	Nagy	projektový manažér
xnovak	Jan	Novák	administrátor
xmala	Jana	Mala	programátorka

Zmazať

Změnit Zrušit

Obrázok 6.6 b) Správa projektu - pracovníci

Na obrázku 6.6 a) je vidieť základné informácie týkajúce sa otvoreného projektu. Sú to informácie, ktoré boli zadané pri jeho vytvorení. Tieto informácie má nárok vidieť aj používateľ bez práv projektového manažéra. Jediným rozdielom je, že ich nemôže zmeniť. V takomto prípade bolo treba skontrolovať role používateľov a podľa toho nastavovať vlastnosť komponentu “Enabled”. V prípade, že používateľ má právo na ich zmenu, povolíme zmenu všetkých vstupných pol pomocou vlastnosti *Enabled = true*. Implicitne nastavená vlastnosť týchto vstupných pol je *Enabled = false*, teda v prípade neoprávneného používateľa netreba previesť programátorsky žiadne zmeny. Po úspešnej aktualizácii údajov sú tieto zmeny hneď premietnuté do hlavného rozhrania systému.

Obrázok 6.6 b) znázorňuje druhú stránku tejto funkcie. Obsahuje dve tabuľky, kde v prvej sú vymenovaní všetci používatelia, kým v druhej, používatelia, ktorí sú priradení na daný projekt. Tieto údaje má právo meniť tak isto ako v predchádzajúcom prípade len projektový manažér, ktorý otvorený projekt vytvoril. Tento manažér môže do tabuľky projektu priradených pracovníkov, zaradiť nových členov tímu, alebo zmazať existujúcich. Systém uskutočňuje kontrolu nad týmito priradeniami, aby nebolo možné pracovníka zaradiť na projekte viackrát. Je kontrolovaná aj duplicita projektových manažérov, čo zabezpečuje aby nebolo možné zaradiť si na projekt dvoch alebo viacerých používateľov s právom projektového manažéra. Táto funkcia systému je implementovaná pomocou objektu typu *DataSet*, kde obidve tabuľky sú inštancie tejto sady dát. Vlastné priradenie pracovníka k projektu, zahŕňa v sebe kopírovanie údajov z jednej sady dát do druhej. Samozrejme je treba počítať aj so zmazanými údajmi a v prípade uloženia tieto údaje zmazať. Kontrola zmazaných údajov bola implementovaná s porovnávaním starej sady dát priradených pracovníkov s novou sadou. V prípade nezhody boli údaje zmazané z tabuľky *Prac_proj*. Táto tabuľka definuje priradenia a je spojovacia medzi tabuľkami *Pracovník* a *Projekt*.

6.3.8 Správa procesov projektu

Procesy projektu sú v tomto systéme definované ako jednotlivé etapy vývoja projektu, v ktorom môžu byť identifikované riziká. V rámci návrhu som sa ešte rozhodol, že systém riziká nebude identifikovať len na úrovni projektov, ale ich bude rozoberať v jednotlivých etapách ich vývoja. Tieto etapy sú v systéme definované ako procesy projektu.

Tieto procesy sú v systéme znázornené ako stromová štruktúra zobrazená na obrázku 6.5 a práca s nimi je popísaná v kapitole 6.3.6.

Pridanie nového procesu

Pridať proces je možné k projektu alebo priamo k nejakému inému procesu. Je to možné preto, lebo procesy sú hierarchicky usporiadané v stromovej štruktúre a môžu byť ľubovoľne zanorené. Keď chceme pridať proces do projektu, musíme vybrať zo stromovej štruktúry ten uzol, do ktorého bude samotný proces priradený. V prípade priradenia priamo k projektu, je treba si vybrať koreňový uzol.

Keď už máme vybrané miesto procesu, stačí kliknúť na ikonu *pridať proces*. Zobrazí sa formulár, kde sú požadované údaje rozdelené na povinné a nepovinné. Prvým povinným údajom je miesto pridaného procesu, čo systém automaticky vyplní podľa predchádzajúceho výberu. Tento údaj je možné zmeniť na hociktorý proces alebo na projekt a tak ovplyvniť miesto pridaného procesu. Okrem tohto údajja je požadované za povinné aj meno a popis. Nepovinnými údajmi sú priorita procesu a náklady na proces, ktoré však nie sú až tak dôležité pre riadenie rizík, len figurujú ako spresňujúce informácie. Systém uskutočňuje pred spracovaním údajov kontrolu, aby nebolo možné formulár poslať bez povinných informácií. Po úspešnej kontrole sú dáta uložené do databázy a je prevádzaná aktualizácia stromovej štruktúry.

Zmena údajov o procese

Zmeniť údaje niektorého procesu projektu je možné obdobným spôsobom, ako zmeniť údaje o projekte. V prípade ak zmeníme výber z projektu na proces, zmení sa názov ikony „podrobnosti o projekte“ na názov „podrobnosti o procese“.

Po kliknutí na ikonu sa zobrazí ten istý formulár ako pri pridávaní, len s vyplnenými údajmi, ktoré je možné meniť. Keď používateľ nebude mať právo projektového manažéra, tieto údaje mu budú tak isto poskytnuté, ale bez možnosti zmeny. Pred uložením samotných zmien, systém prevedie tie isté kontroly ako pred zadávaním, aby sa do databázy nedostali nekonzistentné údaje. Po úspešnej kontrole sú dáta zmenené v databáze a sú aktualizované všetky výpisy týchto údajov v systéme. Tieto zmeny sa uskutočnia priamo v databáze a z týchto dát je potom aktualizovaná odpojená vrstva. Odpojenú vrstvu reprezentuje v systéme dátová sada *RmDataSet*, ktorá je objekt typu *DataSet*.

Zmazanie procesu

Zmazanie procesu je možné uskutočniť po kliknutí na vyznačenú ikonu, ktorú je vidieť na obrázku 6.5 pod stromovou štruktúrou etáp projektu. Keď používateľ klikne na danú ikonu, systém generuje potvrdenie na zmazanie, aby nedošlo k náhodnému činu. Po úspešnom potvrdení systém kontroluje, či k danému procesu sú priradené nejaké riziká. V prípade, že proces riziká obsahuje, používateľ je vyzývaný, aby odstránil najprv tieto riziká a až potom z bezpečnostných dôvodov proces. Systém tak isto nedovoľuje odstrániť proces, ktorý obsahuje podriadené procesy a na túto skutočnosť je aj používateľ upozornený. Ak proces riziko alebo podproces už neobsahuje je bezproblémovo zmazaný z databázy a je odstránený aj zo stromovej štruktúry.

6.3.9 Správa rizík

Do kapitoly správa rizík som začlenil samotnú identifikáciu a uloženie rizika, zmenu údajov a prípadné odstránenie rizík zo systému. Ďalšie funkcie týkajúce sa rizík podrobne popíšem zvlášť v ďalších kapitolách práce.

Pridanie rizika pomocou sprievodcu

Pridanie rizika som sa rozhodol implementovať pomocou sprievodcu pridania, ktorý je používateľom dobre známi postup z aplikácii operačného systému Windows. Dôvodom tohto rozhodnutia bolo, že riziko evidované v systéme obsahuje veľa atribút. Naplniť tieto atribúty v jednom formulári by bolo aj pre skúseneho používateľa ťažké a neprehľadné.

Jednotlivé kroky pridania som sa pokúsil rozdeliť podľa logickej príslušnosti rizikových atribút. Tieto kroky sú nasledujúce:

Krok 1 - v prvom kroku je používateľovi umožnené inšpirovať sa z rizík centrálnej databázy, ktoré je pre jeho identifikované riziko najpodobnejšie. Keď používateľ vyberie riziko z centrálnej databázy, jeho základné údaje budú v ďalšom kroku už vyplnené. Riziko ale nemusí byť vybrané z existujúcej databázy, ale môže byť definované aj od začiatku.

Krok 2 - v druhom kroku systém požiada o základné informácie o riziku. Sú to:

- **názov** – výstižný, skrátený názov rizikovej udalosti,
- **popis** – podrobnejší popis rizikovej udalosti,
- **kategória** – dopredu definované kategórie, do ktorých riziková udalosť spadá,
- **typ** – typ rizika v rámci kategórie spravovaní projektovým manažérom.

Kategóriu je možné vybrať si z rolovacieho menu a jeho hodnoty sú fixne dané a boli prebrané z literatúry [1]. Typy rizík zase má na starosti projektový manažér a je možné pridať nový typ priamo zo sprievodcu pridania rizika. Názov a popis sú vstupné pole, ktoré musia byť korektne vyplnené, aby bolo možné postúpiť k tretiemu kroku.

Krok 3 - v tomto kroku je možné definovať k riziku vlastníka a sú zobrazené aj údaje o identifikujúcom, ktorý je analogicky prihlásení pracovník. Priradiť vlastníka k riziku nie je nevyhnutné, je možné to urobiť neskôr pri zmene údajov.

Krok 4 - štvrtý krok obsahuje identifikátory rizikovej udalosti. Sú to:

- **stav** – treba zadať v akom stave je momentálne riziková udalosť,
- **príčina** – zadáva sa prvotná príčina rizikovej udalosti,
- **spúšťač** – uvedú sa indikátory skutočného vzniku rizikovej udalosti.

Príčina a spúšťač sú vstupné pole do ktorých je treba zadať požiadané údaje. Stav je komponenta typu rolovacieho menu a má fixne dané hodnoty. Tieto hodnoty udávajú v jakom stave je momentálne riziko ktoré môžu byť hodnoty: koncept, aktívne a uzavreté.

Krok 5 - v piatom kroku používateľ musí zadať dve hodnoty:

- **pravdepodobnosť** – pravdepodobnosť vzniku rizikovej udalosti,
- **dopad** – aký veľký bude dopad pri vzniku rizikovej udalosti.

Tieto hodnoty je možné zadať podľa tabuľky hodnotenia, ktoré sú zobrazené v spodnej časti formulára. Tabuľky hodnotenia ako pre pravdepodobnosť, tak pre dopad môžu zadefinovať identifikujúci rizika samostatne podľa ich predstáv. Priamo je to možné teraz pri identifikácii, alebo pri neskorších úpravách rizika. Okrem nadefinovania nových hodnôt je možné priradiť hodnotenie

aj z existujúcich, už v systéme registrovaných hodnotení. Z hodnoty pravdepodobnosti a dopadu je potom vypočítaná závažnosť rizika, na ktorej je najväčší dôraz pri riadení a rozhodovaní. Tieto hodnoty musia byť korektne vypočítané pre správne manažérske rozhodnutia.

Krok 6 - šiesty a ostatné kroky sprievodcu už nie sú povinnými pre identifikovanie, ale špecifikujú a spresňujú údaje o riziku. Tieto údaje bývajú vyplnené hlavne až po identifikácii, ale v niektorých prípadoch sa už zadávajú pri registrácii rizika. Šiesty krok sprievodcu umožňuje zadávanie reakcií na riziko, ktoré pridávame. Tieto reakcie je možné zadať priamo, alebo priradiť už z existujúcich reakcií. Zo správou a aktualizáciou týchto reakcií sú poverení projektoví manažéri.

Krok 7 - je posledným krokom, ktorý umožňuje priradenie súborov k riziku hneď pri identifikácii. V praxi sem často dávajú zápisnice z konferencií, kde bolo riziko identifikované, alebo prípadné sprievodné grafy. Ku každému riziku je možné priradiť súbory s príponou .doc, .xls, .pdf a .txt. Súbory patriace k rizikám budú detailne popísané v kapitole uvedenej ďalej.

Zmena údajov o riziku

Údaje o rizikách je možné zmeniť priamo pri výpise podrobných informácií. Tieto informácie sú vypísané do viacstránkového okna, medzi ktorými je používateľovi umožnené jednoduché pohybovanie.

Údaje na stránkach sú rozdelené podľa logickej príslušnosti, podobne ako pri sprievodcu pridávania rizika. Keď je do systému prihlásený projektový manažér otvoreného projektu, alebo vlastník rizika, sú mu tieto údaje poskytnuté na zmenu. V ostatných prípadoch je možné na tieto údaje len pozrieť, vytlačiť z neho kartu rizika alebo previesť XML export. Tak ako je vidieť na obrázku 6.7 zmeniť je možné všetky atribúty rizika. Okrem zmeny je možné priradiť alebo zmazať reakciu, zadať nové hodnotenia pravdepodobnosti a dopadu, priradiť nové súbory, prípadne zmazať staré.

Systém automaticky monitoruje aj činnosť používateľa a v prípade potreby ponúkne aj možnosť uloženia týchto údajov do databázy. Keďže používateľ môže prevedené zmeny hocikedy zrušiť, bolo treba systém implementovať tak, aby hodnoty neboli hneď zapísané do databázy len po potvrdení používateľom. Aj v takýchto prípadoch je využitie odpojenej vrstvy ADO.NET efektívnejšie a ľahšie na programovanie. Stačí všetky zmeny uložiť do dočasnej dátovej sady a v prípade koncového uloženia reflektovať do originálnej databázy. Keď používateľ sa rozhodne prevedené zmeny zrušiť, dočasná dátová sada sa do databázy neuloží a všetky údaje zostanú nezmenené.

Pravděpodobnost	Hodnota	Dopad	Hodnota
Nepatrná	0-20	Nepatrný	0-20
Malá	21-40	Malý	21-40
Střední	41-60	Čekný	41-60
Velká	61-80	Kritický	61-80
Mimořádná	81-100	Katastrofický	81-100

Název hodnocení	od	do
Nepatrná		
Malá		
Střední		
Velká		
Mimořádná		

Obrázok 6.7: Zmena údajov o riziku

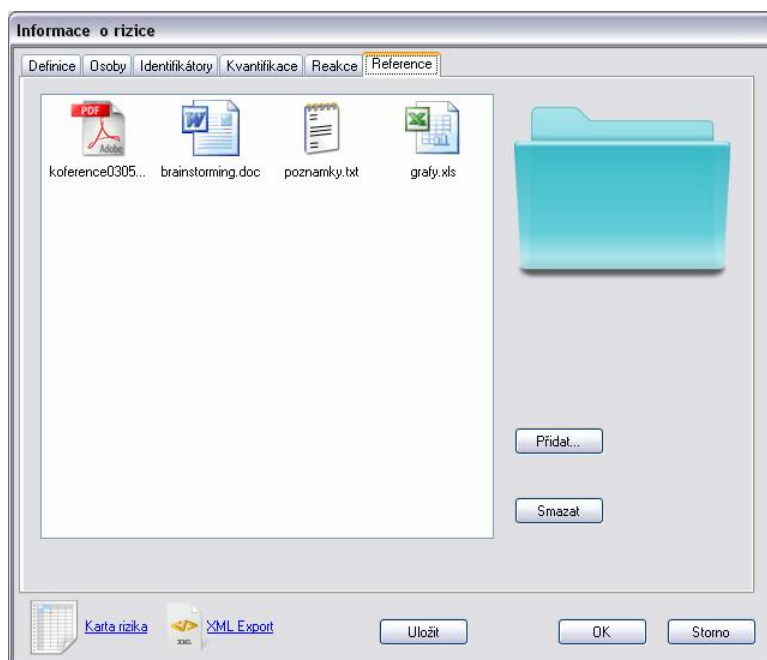
Zmazanie rizika

Zmazať riziká sa dá jednoducho kliknutím na príslušnú ikonu. Táto ikona je aktívna len v prípade prihláseného projektového manažéra otvoreného projektu alebo pri vlastníkovi rizika. Z tejto skutočnosti vyplýva, že ostatní používatelia nemajú právo riziko zo systému zmazať.

Akurát používateľ klikne na ikonu zmazania rizika, je vyvolané bezpečnostné preverenie odhaľujúce náhodné činy používateľa. Riziko je povolené na zmazanie v každom prípade, netreba teda zvlášť zmazať jeho referencie, tie budú spolu zmazané s ním. Zmazanie jedného rizika zahŕňa v sebe zmazanie aj iných hodnôt z databázy. Aby po zmazaní nezostali v systéme takzvané „smeti“, je treba tieto hodnoty v každom prípade efektívne odstrániť. Hodnoty, ktoré budú spolu s rizikom zmazané sú napr.: súbory, riadok spojovacej tabuľky *Proces_riziko* a *Riziko_reakcia* a časové hodnoty v jednotlivých mesiacoch. Keby tieto hodnoty neboli zmazané spolu s rizikami, obsadzovali by zbytočne miesto v databáze a pritom použité by neboli nikdy. Po úspešnom zmazaní je riziko a jeho referencie odstránené z databázy a register rizík aktualizovaný.

6.3.10 Priradenie súborov k rizikám

K rizikám okrem iných atribút, ktoré boli vymenované v kapitole 6.3.9, je možné priradiť aj referencie na súbory, čo je znázornené na obrázku 6.8.



Obrázok 6.8: K riziku priradené súbory

Tieto súbory môžu obsahovať spresňujúce informácie o rizikách, prípadné sprievodné tabuľky, grafy. Pridané referencie môžu byť súbory obsahujúce prípony:

- **.doc** – súbory vytvorené v programu Microsoft Word,
- **.xls** – súbory vytvorené v programu Microsoft Excel,
- **.pdf** – súborový formát vyvinutý firmou Adobe,
- **.txt** – súborový formát vyvinutý firmou Microsoft.

Preto som sa rozhodol implementovať pridávanie týchto súborov, lebo sú najviac využívané pre uchovávanie informácií v operačných systémoch Microsoft Windows. V prípade iných súborov je určite možné bezproblémové prevedenie do jedného z vymenovaných formátov.

Súbory je možné priradiť priamo pri pridávaní, alebo pri neskorších úpravách rizika. Práve tieto úpravy je vidieť na obrázku 6.7, kde sú vymenované podrobné informácie o riziku spolu s referenciami na súbory. Keď používateľ klikne na tlačidlo pridať zobrazí sa mu dialógové okno, kde si môže vybrať súbor na pridanie. Toto dialógové okno je implementované pomocou komponentu *OpenFileDialog*. Bolo treba používateľa zamedziť v tom, aby do systému zadával súbory, ktoré nie sú podporované. Vyriešil som to pomocou filtra, ktorý je jeden z najužitočnejších vlastností komponentu *OpenFileDialog*. Do filtra je možné zadať prípony súborov, ktoré potom používateľ bude môcť otvárať cez túto komponentu. Po vybraní je daný súbor prekopírovaný do zložky, ktorá bude obsahovať všetky súbory zoskupené podľa rizík. Tieto súbory je potom možné priamo z programu spúšťať a zmeniť ich obsah. Do problémov som sa dostal práve z tej skutočnosti, že používateľ môže svoje zmeny hocikedy zrušiť a vtedy je treba všetky zmeny tak isto anulovať. Práve preto bola potrebná dôkladná kontrola na stav databázy pred samotnými zmenami a na stav po zmenách. Keď používateľ vymazal niektorý súbor, ojazstné vymazávanie bolo treba urobiť až po potvrdení týchto zmien. Keby to nebolo implementované takýmto spôsobom, mohlo by dôjsť k stavu nekonzistencie databázy. Všetky úpravy so súbormi sú implementované pomocou odpojenej vrstvy ADO.NET a reflektované do databázy až po potvrdení týchto úprav používateľom.

Počet priradených súborov k riziku nie je žiadnym spôsobom limitované, jediným obmedzením je veľkosť voľného miesta na disku, kde je program sprevádzkovaný. Všetky úpravy zo súbormi sú prevedené na kópie a v prípade zmazania je odstránená táto kópia, v žiadnom prípade nie originálny súbor.

6.3.11 Kvalitatívne hodnotenie a časový graf

Kvalitatívne hodnoty reprezentujú v systéme pravdepodobnosť a dopad rizikovej udalosti. Podľa týchto hodnôt je vypočítané riziko, ktoré jednoznačne udáva závažnosť identifikovanej udalosti.

Pravdepodobnosť a dopad sa udáva v celočíselnej podobe a pre tieto hodnoty sa zvlášť definuje aj tabuľka hodnotenia. Podľa týchto hodnotení sú pravdepodobnosti a dopady rozdelené do cieľových skupín. To, že aká bude pravdepodobnosť a dopad hodnotení, sa udáva pri identifikácii rizika, ale môže byť v rámci vývoja projektu hocikedy zmenený. Pri definovaní jednotlivých rozmedzí sa musí udávať dolná a horná hranica hodnôt a tieto hodnoty musia byť rozdelené do piatich skupín. Predpokladá sa rozdelenie podľa vhodne zvolenej distribučnej funkcie, čo je konvertované do kvalitatívnej a kvantitatívnej škály. Systém automaticky kontroluje logické chyby pridávaných intervalov.

Príkladom by mohla byť tabuľka 6.1, ktorá rozdeľuje limitný interval parametra pravdepodobnosti (v tomto prípade od 1 do 100).

Intervaly hodnôt pravdepodobnosti	Kvalitatívna stupnica	Kvantitatívna stupnica
1-20	Nepatrná	1
21-40	Malá	2
41-60	Stredná	3
61-80	Veľká	4
81-100	Mimoriadna	5

Tabuľka 6.1: Tabuľka hodnotenia pravdepodobnosti výskytu udalosti

Tak isto ako v predchádzajúcom príklade je možné uviesť aj tabuľku 6.2, ktorá rozdeľuje limitný interval parametru dopadu (v tomto príklade od 1 do 50).

Intervaly hodnôt dopadu	Kvalitatívna stupnica	Kvantitatívna stupnica
1-10	Nepatrný	1
11-20	Malý	2
21-30	Citeľný	3
31-40	Kritický	4
41-50	Katastrofický	5

Tabuľka 6.2: Tabuľka hodnotenia dopadu na sledovaný parameter

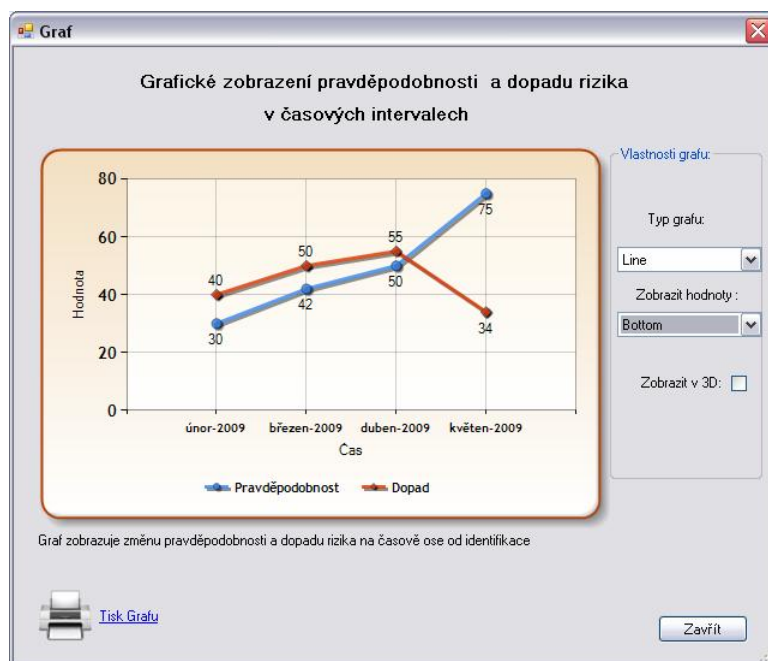
Výsledné riziko potom vypočítam tak, že vynásobím kvantitatívnu stupnicu pravdepodobnosti s kvantitatívnou stupnicou dopadu identifikovaného rizika. Nakoniec k týmto výsledným numerickým hodnotám priradím kvalitatívne hodnotenie. Výsledok odvodený z tabuľky 6.1 a z tabuľky 6.2 by mohol vyzeráť tak, ako je to znázornené v tabuľke 6.3.

		Pravdepodobnosť				
	$R=(1) \times (2)$	1: Nepatrná	2: Malá	3: Stredná	4: Veľká	5: Mimoriadna
Dopad	1: Nepatrný	1:Nepatrné	2:Nepatrné	3:Nepatrné	4:Malé	5:Malé
	2: Malý	2:Nepatrné	4: Malé	6: Malé	8:Stredné	10:Stredné
	3: Citeľný	3:Nepatrné	6: Malé	9: Stredné	12: Stredné	15:Veľké
	4: Kritický	4: Malé	8: Stredné	12: Stredné	16: Veľké	20:Mimoriadne
	5: Katastrofický	5: Malé	10: Stredné	15: Veľké	20:Mimoriadne	25:Mimoriadne

Tabuľka 6.3: Výsledná tabuľka rizík (odvodené z Tabuľky 6.1 a Tabuľky 6.2)

Tabuľka vyjadruje, že keď do systému vstupuje riziko z malou začiatkovou kvantitatívnou hodnotou pravdepodobnosti a kritickou kvantitatívnou hodnotou dopadu, samotné riziko sa musí rovnať hodnote strednej.

Hodnota pravdepodobnosti a dopadu je vlastne premenným atribútom rizika, pretože ich hodnoty sa môžu meniť v jednotlivých časových okamžikoch. Pre lepšie riadenie a využívanie rizík som sa rozhodol systém vytvoriť tak, aby každý mesiac uchovával zmenu týchto hodnôt. Keď používateľ vypíše podrobné informácie o riziku, systém mu umožní vypísať si aj časový graf kvantitatívnych hodnôt. Výpis podrobných informácií je na takom istom mieste ako zmena týchto údajov, čo je vidieť vyššie na obrázku 6.7, kde je znázornená aj ikona časového grafu. Po kliknutí sa zobrazí časový graf, ktorý je zobrazený na obrázku 6.9.



Obrázok 6.9: Časový graf pravdepodobnosti a dopadu

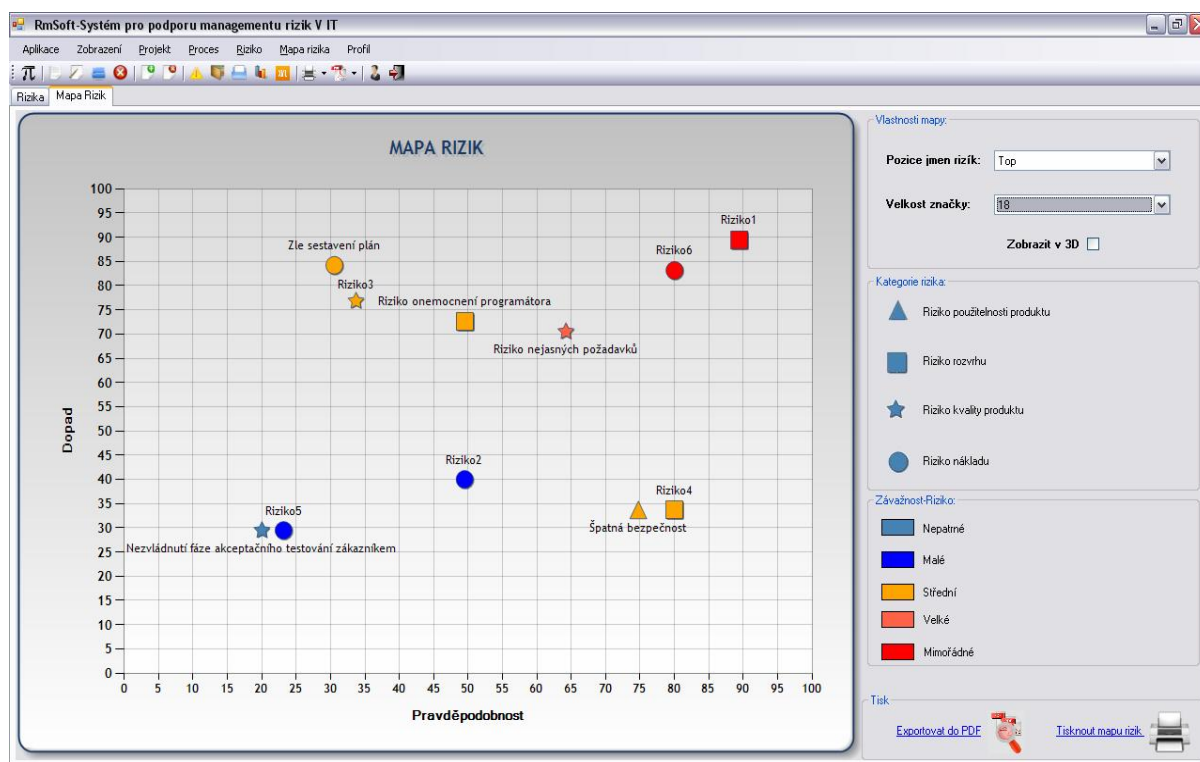
Uchovávanie týchto údajov sa začína od identifikácie rizika až do existencie projektu, alebo do zmazania rizika. Na obrázku 6.9 je vidieť priebeh ako sa hodnoty pravdepodobnosti a dopadu zmenili každý mesiac. Je vidieť, že vo februári, keď riziko bolo identifikované, hodnota pravdepodobnosti sa rovnala 30, kým po troch mesiacoch sa táto hodnota vyšplhala na 75. Podobne sa aj dopad tejto rizikovej udalosti každý mesiac zvyšoval, ale pritom v máji sa podarilo túto hodnotu znížiť na 34, čo je najmenšia hodnota počas existencie rizika. Táto funkcia systému umožňuje projektovým manažérom mať prehľad o tom, ako sa hodnoty pravdepodobnosti a dopadu vyvíjali počas životnosti rizika. Majú prehľad aj o tom, či protiopatrenia, ktoré boli proti riziku prevedené doniesli svoj úspech, alebo naopak zlyhali. Môže tento graf spolu z ostatnými údajmi slúžiť aj ako súvaha o rizikách pre vedenie firmy. Tieto grafy prehľadne odzrkadľujú, ako funguje efektívne riadenie rizík v organizácii. Graf je možné zobrazit' aj v 3D, a pritom je možné zmenit' aj jeho vlastnosti, ako je typ zobrazenie hodnôt a typ línií.

Na úrovni implementácie je časový graf vyriešený tak, že v prípade zmeny hodnôt, je kontrolovaný časový údaj poslednej uloženej hodnoty. Keď tento dátum je zhodný s aktuálnym mesiacom, stará hodnota sa aktualizuje v databáze. V prípade, že sme v inom mesiaci, nová hodnota sa uloží do databázy s novou časovou etiketou. Z tohto vyplýva, že aktuálna hodnota reprezentujúca mesiac, je vždy posledná zadaná hodnota v danom mesiaci. Teda v databáze je uložená časová hodnota ku každému mesiacu len jedna. Keď počítame s tým, že riziká evidované v systéme budú mať priemerne ročnú životnosť, môže nadobúdať v tejto tabuľke pre každé riziko maximálne 12 hodnôt.

6.3.12 Mapa rizík

Riziká, ktoré boli identifikované a registrované v systéme, je možné ďalej skúmať pre každý projekt zvlášť v tzv. mape rizík. Mapa rizík je dokladom, ktorý prostredníctvom grafického vyjadrenia, alebo tabuľkového prehľadu rizikových faktorov, poskytuje informácie o stupni významnosti rizík z hľadiska ich nežiaduceho dopadu a pravdepodobnosti výskytu.

V systéme je implementovaná mapa rizika ako dvojrozmerný graf polôh rizík v súradniciach pravdepodobnosť a dopad rizika. Tento graf je dostupný pre každý projekt zvlášť ako druhá stránka hlavnej aplikácie vedľa registra rizík. Každá mapa rizika vyzerá inak a závisí od počtu, kategórií a závažnosti rizík. Na obrázku 6.10 je vidieť mapu rizika pre testovací projekt, na ktorý bolo identifikovaných 11 rizík.



Obrázok 6.10: Mapa rizík

V grafe sú jednotlivé riziká zoskupené podľa kategórie a závažnosti. To, do akej kategórie riziko spadá definuje jeho tvar, kým závažnosť rizika udáva jeho farba. Analogicky závažnosť rizika bude udávať aj jeho pozícia v grafe. Riziká, ktoré sú bližšie k ľavému dolnému rohu grafu, sú riziká s menšou závažnosťou, kým riziká, ktoré sa blížia k pravému hornému rohu nadobúdajú väčšie závažnosti. Legenda na pravej strane tejto mapy definuje, ako sú zobrazené v mape kategórie a závažnosti rizík. Na obrázku 6.10 je vidieť, že Riziko 1 má formu červeného štvorca, čo znamená že spadá do kategórie nákladov a jeho závažnosť je mimoriadna. V mape je možné nastavovať pomocou roletového menu pozíciu názvu rizika a veľkosť značky zobrazujúceho riziko. Zobrazenie celého grafu je možné prepnúť aj do 3D modelu, pre lepšie vyobrazenie hodnôt rizík.

Súradnice mapy sú v každom prípade konštantne rozdelené od 0 do 100. Keďže pravdepodobnosť a dopad rizika môže byť ohodnotená u každého rizika s inou stupnicou, tieto stupnice bolo treba prekonvertovať na ekvivalentný formát. Toto konvertovanie prebieha priamo v rámci vykreslenia grafu. Funkcia si vyberie všetky riziká, ktoré patria k danému projektu a jednotlivito vypočíta ich umiestnenie v mape, podľa aktuálnej hodnoty a aktuálneho hodnotenia.

Keď sa projektový manažér pozrie na túto mapu hneď mu musí byť jasné, ktoré riziká ohrozujú najviac a ktoré sú nepatrné pre dokončenie projektu. Výsledky môže prekonvertovať do PDF súboru, alebo vytlačiť cez tlačiareň. Systém pre oba voľby vygeneruje ukážku pred tlačou, v ktorom sú okrem mapy obsadené aj informácie o projekte. Tieto výstupy môžu byť využité ako súhrnné správy kontroly riadenia rizík na daný projekt. Mapa rizika je okamžite aktualizovaná, akurát je registrované nové riziko k projektu, alebo sa menia hodnoty pravdepodobnosti a dopadu rizík.

6.3.13 Export / Import rizík

Riziká sú najdôležitejšími entitami celého systému, na ktorých je stavená celá implementovaná aplikácia aj jeho funkcie. Z tohto dôvodu bolo potrebné zabezpečiť, aby údaje o rizikách mohli byť prenášané medzi rôznymi projektmi, aplikáciami a platformami. Rozhodol som sa preto do systému implementovať funkciu na export týchto údajov do formátu XML.

Export a Import do XML

XML je formát súboru obsahujúci dáta. Zvláštnym ho robí to, že je široko prijímaný ako štandard so schopnosťou existencie na akejkolvek platforme a používaný masívnym, stále rastúcim počtom aplikácií a programovacích jazykov. Programovací jazyk C# s technológiou .NET, v ktorom bol môj systém vytvorený, plne podporuje tento formát a ponúka aj rôzne zabudované funkcie pre ľahšiu prácu s ním. Pomocou XML je možné vymieňať dáta aj medzi dvoma úplne odlišnými systémami.

Exportovať do súboru XML je možné každé riziko zvlášť. Ikona na export sa nachádza v podrobných informáciách rizika, alebo priamo v nástrojovej lište. Po výbere je používateľ vyzvaný na zadanie mena a cesty XML súboru, kde budú vyexportované tieto údaje. Pre výber súboru a cesty som používal komponentu *SaveFileDialog*. Táto komponenta obsahuje vlastnosť *Filter*, pomocou čoho je možné zadať typ uloženého súboru. Exportovaná do súboru nie je len samotná tabuľka rizika, ale aj iné tabuľky, ktoré sú s ním spojené. Znamená to, že v prípade exportu spolu s rizikom budú vyexportované aj všetky údaje o vlastníčkovi, časových hodnôt, reakcií atď.. Tieto údaje nie je potrebné jednotlivo zapísať do textového súboru pomocou pravidiel XML, ale stačí používať funkciu *WriteXml* objektu typu sady dát. Táto funkcia musí byť volaná na platnú dátovú sadu (*DataSet*), z ktorej budú dáta vyexportované. Funkcia potrebuje ako vstupný parameter názov súboru, do ktorého bude zapisovať.

Ako prvý krok som teda musel z rizika a z ostatných tabuliek, ktoré sú s ním spojené, vytvoriť novú dátovú sadu. Keď som už úspešne vytvoril túto dátovú sadu, stačilo zavolať na tento objekt *WriteXml* a poskytnúť mu hodnotu získanú z komponentu *SaveFileDialog*. Takto vytvorený XML súbor bude obsahovať všetky potrebné údaje o riziku, ktoré je potom možné naimportovať späť do iného procesu, projektu alebo do úplne inej aplikácie.

Import údajov je možné previesť z hlavného rozhrania kliknutím na ikonu „Import riziko z XML“. Ešte však pred samotným importom musí byť vybraný proces, do ktorého bude riziko priradené. Import do dátovej sady je prevedení podobným spôsobom ako Export a slúži na to funkcia objektu typu sady dát *ReadXml*. Táto funkcia vráti ako výsledok objekt typu *DataSet*, v ktorom sú uložené načítané dáta. Ako vstupný údaj však potrebuje samotný XML súbor, z ktorého načítanie môže previesť. Meno a cesta k cieľovému XML súboru je získaná pomocou komponenty *OpenFileDialog*. Je to dobre známa komponenta z operačných systémov Windows, pomocou ktorého je možné vybrať akýkoľvek súbor.

Údaje, ktoré som načítal z XML súboru a uložil do dátovej sady, bolo treba uložiť aj do databázy. Aby nedošlo k duplicite dát, bolo treba kontrolovať pridávanie týchto hodnôt. Keďže tieto hodnoty nie je možné porovnávať podľa jedinečných identifikátorov (podľa ID), porovnával som ich podľa ostatných atribút. V prípade, že všetky atribúty boli rovnaké, predpokladal som že načítaný údaj je totožný s nájdeným a priradil som ho k načítanému riziku.

Teda aplikácia funguje tak, že po načítaní dát z XML súboru sa pokúsi k týmto dátam nájsť ekvivalentné hodnoty z aktuálnej databázy. Keď nájde ekvivalentnú hodnotu v databáze, priradí to k načítanému riziku. V opačnom prípade nová hodnota sa uloží do databázy a je podľa nového identifikátora priradená k načítanému riziku. Systém si teda poradí aj s takými hodnotami, ktoré

v databáze nie sú uložené, ale aj s takými, ktoré sa v ňom nachádzajú. Takýto typ exportu a importu by nebol problémom rozšíriť aj na úroveň procesov a projektov.

Export do PDF

Okrem exportu do XML súboru systém umožňuje exportovať údaje aj do PDF súboru. Exportovať do PDF je možné kartu rizika a mapu rizika, ktoré obsahujú aj informácie o projekte. Je to dobré z hľadiska zdieľania informácií medzi jednotlivými členmi projektu, alebo vedúcimi pracovníkmi firmy v otvorenom formáte.

Export údajov do PDF súboru je v celom systéme implementované pomocou voľne dostupnej knižnice *iTextSharp*. *iTextSharp* je odvodení z pôvodnej knižnice *iText*, ktorý je používaní v Jave. Vývoj sa stále odohráva v Jave, ale k jednotlivým verziám sa vydávajú aj porty na .NET. Táto knižnica je dostupná na adrese uvedenej v literatúre [17]. Samotná tlač textu do PDF je možná pomocou *iTextSharp*. Jednoducho stačí používať objekt typu *iTextSharp.text.Paragraph*, ktorý potrebuje ako vstupný parameter exportovaný text. Tento text je potrebné potom pridať do dokumentu, ktorý bol vytvorení na začiatku ako inštancia objektu *Document*. Graf exportovať už nebolo také ľahké, lebo ho bolo potrebné najprv prekonvertovať do formátu *Bitmap* a potom ako objekt *System.IO.MemoryStream* do formátu .png. Keď už graf bol prekonvertovaný na potrebný formát, stačilo ho priradiť k objektu typu *iTextSharp.text.Image* a pridať ho do vytvoreného dokumentu.

6.3.14 Karta rizika

Karta rizika slúži projektovému manažérovi alebo vlastníkovi rizika pre zobrazenie základných informácií o riziku a pritom ako upresňujúci podklad k pravidelnej správe o riziku. Obsahuje všetky dôležité sledované údaje o riziku, spolu s priradenými reakciami a prílohami. Príklad takejto karty rizika je vidieť na obrázku 6.11.

Obrázok 6.11: Karta rizika

Kartu riziku môže vygenerovať zo systému každý registrovaný používateľ, ktorý nie je viazaný na žiadne bezpečnostné právo. Ikona na generovanie je dostupná priamo z nástrojovej lišty, z menu alebo z výpisu podrobných vlastností rizika. Údaje, ktoré sú zobrazené na karte, sú zjednodušené formy podrobných popisov rizika. Znáročujú len dôležité a neodpušiteľné informácie o riziku pre štatistické súhrny.

Na začiatku sú popísané len základné informácie, ako sú názov, popis, kategorizácia, vlastník a používateľ. Pracovník, ktorý riziko identifikoval je uvedený spolu s dátumom identifikácie. Po základných údajoch nasleduje podrobný popis prvotnej príčiny vzniku rizikovej udalosti a zobrazujú sa aj indikátory skutočného vzniku. Z karty rizika nesmie chýbať ani grafické zobrazenie, do akej kategórie spadá odhad pravdepodobnosti vzniku a odhad dopadu v prípade skutočného vzniku. Na konci karty sú uvedené mená reakcií a priložených súborov. Aby som kartu rizika zbytočne nekomplikoval, podrobné popisy reakcií som vynechal. Výstižný názov musí byť dostačujúci pre štatistický súhrn týchto údajov. Systém kartu umožňuje aj vytlačiť cez tlačiareň, alebo priamo vyexportovať do PDF súboru, s postupom, ktorý bol popísaný v predchádzajúcej kapitole 6.3.13.

6.3.15 Zmena vlastných údajov používateľa

Základnou funkciou každého moderného informačného systému je aj zmena vlastných údajov prihláseného používateľa. Profil používateľa je dostupný z hlavného rozhrania, alebo v každom prípade z nástrojovej lišty a menu. Po výbere systém poskytne formulár s vyplnenými údajmi používateľa, ktorý je vidieť na obrázku 6.12.

Profil přihlášeného uživatele

Profil uživatele

Údaje které můžete změnit:

Jméno: Petr

Příjmení: Nagy

Login: xnagy

Email: petr.nagy@seznam.cz

Heslo: xxxxxx

Potvrdit heslo: xxxxxx

Když nechcete změnit heslo, pole nechejte prázdné!

Údaje které může změnit jenom Váš administrátor systému:

Funkce: projektový manažér

Nadřazení: není

Právo v systému jako: Projektový manažér

Uložit Zavřít

Obrázok 6.12: Údaje prihláseného používateľa

Na obrázku 6.12 je vidieť, že formulár je rozdelený na dve časti. Prvá časť obsahuje údaje používateľa, ktoré môže zmeniť bez obmedzenia. Druhá časť mu zobrazuje údaje, ktoré nemôže zmeniť. Na zmenu týchto údajov má právo len administrátor systému, ktorého je potrebné v takomto prípade požiadať.

V prípade, keď používateľ nechce zmeniť svoje heslo, toto pole môže nechať prázdne. Systém bol implementovaný tak, aby pred samotnou zmenou kontroloval identitu prihlasovacieho mena. Znamená to, že používateľ nemôže zmeniť svoje prihlasovacie meno na také isté meno, ktoré je už v systéme zaregistrované. V prípade identických prihlasovacích mien, by systém nevedel bezpečne

autentifikovať používateľa pri prihlásení. Tak isto z bezpečnostných dôvodov je používateľ požiadaný na zadanie hesla dvakrát. V prípade, že toto opatrenie by nebolo v systéme implementované, používateľ by mohol svoje heslo pokaziť s preklepom a nabudúce by sa do systému nevedel prihlásiť. Preto, používateľ keď zadá svoje heslo ešte raz, je veľmi malá šanca na to, že svoje heslo pokazi dvakrát na takom istom mieste.

Po úspešnej kontrole systém zostrojí dotaz *UPDATE* na používateľa identifikovaného podľa *ID*, pomocou ktorého zmení údaje. Zmenené údaje budú v systéme hneď aktívne a používateľovi sa netreba zvlášť odhlasovať a prihlasovať.

6.3.16 Správa ďalších údajov týkajúce sa rizík

Nad správou ďalších údajov týkajúcich sa rizík sa rozumie správa reakcií, typov a hodnotení. Tieto údaje je možné pri pridávaní rizika zadať alebo priradiť z existujúcich. Systém by mal ale tieto údaje poskytnúť aj na zmenu, prípadne na zmazanie. Pod týmito funkciami sa rozumie v tejto kapitole správa ďalších údajov. Správu týchto údajov budú mať na starosti projektoví manažéri systému.

Správa týchto údajov je na každej úrovni implementácie stavená na podobný princíp. Znamená to, že všetky tieto funkcie budú mať podobné grafické používateľské rozhranie a podobnú aplikačnú logiku. Medzi obrázkami 6.13 a) a 6.13 b) je vidieť podobnosť správy týchto funkcií.

Obrázok 6.13 a) Správa hodnotení dopadu

Obrázok 6.13 b) Správa reakcií na riziká

Projektovému manažérovi systému si stačí vybrať jedno hodnotenie, reakciu alebo typ z tabuľky vymenovaných hodnôt a potom môže poskytnuté atribúty ľubovoľne zmeniť. Jediným obmedzením zmeny týchto údajov je to, aby zmenený údaj nebol viazaný k žiadnemu riziku. V prípade pokusu o takúto zmenu systém vypíše varovné hlásenie.

Táto kapitola sa zaoberala vlastnou implementáciou systému pre riadenie rizík v IT. Na začiatku kapitoly som rozpísal samotné implementačné nástroje a systémy, ktoré som používal pri vývoji. Nasledovala samotná implementácia databázových tabuliek a dvojaký prístup k týmto dátam. Na koniec kapitoly som rozpísal aj hlavné funkcie systému a demonštroval som ich s pridanými fotografiami. V niektorých prípadoch som sa snažil popísať aj použité techniky, riešenia a problémy s ktorými som sa stretol pri vývoji. U jednotlivých funkcií systému som uviedol aj potrebné používateľské práva na ich plnohodnotné využitie. Celú kapitolu som napísal štruktúrované, podľa toho ako som celý systém vyvíjal.

7 Testovanie

Testovanie tvorí významnú časť každého projektu vývoja a implementácie softvéru. Nepredstavuje len jednoduchú a automatizovanú činnosť, ale je to aj dlhodobý proces, pomocou ktorého dosahujeme požadovanú kvalitu softvéru z hľadiska funkčnosti, spoľahlivosti, výkonnosti, použiteľnosti a podporovateľnosti. Pri enormnom tempe v oblasti vývoja softvéru sa aj samotné testovanie stalo netriviálnym procesom zasahujúcim do všetkých etáp životného cyklu softvéru. V mnohých prípadoch predstavuje až polovicu celkového času potrebného na vývoj informačného systému.

Pri testovaní softwaru som použil nasledovné dve skupiny testovania:

- **"black box testovanie"** - tester vie čo má testovaný software robiť, ale nevie ako pracuje vo vnútri,
- **"white box testovanie"** - tester má prístup k obsahu programu (zdrojovému kódu) a vie ako program pracuje.

White box testovanie som robil sám priebežne pri dokončení každej fázy funkcionality systému a chyby som aj hneď opravil. Pokúsil som sa stanoviť dáta pre testovacie prípady na základe znalosti vnútornej logiky programu. Cieľom bolo vykonať všetky príkazy programu tak, aby sa zistili chyby v riadiacej logike. Preto každú možnosť funkcionality som otestoval aspoň raz, čo by malo byť dostačujúce na odhalenie všetkých chýb.

Pri testovaní pomocou postupu „white box“ boli zistené hlavne chyby týkajúce sa aplikačnej logiky systému a chyby pri práci s databázou. Zistil som chyby, ktoré som urobil pri implementácii algoritmov a pri ošetrovaní používateľských vstupov. Vážne chyby boli odhalené aj pri programových konštrukciách, ktoré boli kopírované s úpravami do iných častí programu. Pri takomto spôsobe sa vždy vyskytnú nejaké chyby, ktoré môžu byť hlavne pri prepisovaní premenných, funkcií alebo algoritmov. Častou príčinou existencie chýb bola aj moja nepozornosť pri zostavovaní SQL dotazov a databázových konštrukcií.

Black box testovanie bolo prevedené na používateľoch, ktorí na systém pozerali ako na čiernu skrinku. Ich funkciou bolo, aby zistili, za akých okolností sa chová systém inak, ako bolo uvedené v špecifikáciách. Tieto testy robili používatelia s rôznymi počítačovými znalosťami, preto boli predpokladané u niektorých prípadoch extrémne testovacie hodnoty. V tomto prípade bola testovaná aj použiteľnosť, lebo systém by mal umožniť i menej skúseným používateľom bez väčších problémov s ním pracovať.

Chyby, ktoré boli zistené pri postupe „black box“ testovania, sa týkali hlavne grafických zobrazení a prvkov. Mnohokrát sa stalo, že po vložení nových údajov do databázy, sa tieto údaje nezobrazovali hneď v systéme, len po reštartovaní. Dlhो nezisteným problémom bola aj nesprávne vytvorená mapa rizík. Údaje, ktoré boli prekonvertované z iných hodnotení na ekvivalentný formát boli nesprávne vypočítané, pretože vzorec na výpočet nebol zostavený správne. Okrem grafických zobrazení sa stále vyskytli aj problémy s nedostatočnými ošetrovaniami používateľských vstupov a prípadné nekonzistencie v databáze.

V rámci testovania a pri bežnej práci s aplikáciou som prišiel na to, že sa v kóde môže vyskytnúť veľmi veľa chýb ohrozujúcich správnu funkčnosť aplikácie. Tieto chyby môžu byť kvôli nepresnej komunikácii programátora pri množstve situácií, kedy program nemusí pracovať korektne. Testovanie by malo tieto chyby objaviť, preto je dobré si najprv vytvoriť detailný plán testovania a až potom previesť dôkladné metodické otestovanie funkcií systému.

8 Ďalší rozvoj systému

Pri samotnom vývoji, testovaní a pri nasadení aplikácie do reálneho prostredia, vzniknú nové nápady na rozšírenia, vylepšenia systému. Táto kapitola sa zaoberá práve s týmito nápismi a rozšíreniami systému do budúcnosti, eventuálne aj novými vlastnosťami.

Prvým dôležitým rozšírením by mohla byť identifikácia pravdepodobnosti a dopadu pomocou dotazníkového prieskumu priamo cez systém. Znamenalo by to, že projektoví manažéri by ku každému riziku mohli založiť meranie týchto veličín. Každé meranie by obsahovalo otázky zvlášť na dopad, zvlášť na pravdepodobnosť, ktoré by potom vyplňovali priradení pracovníci. Projektoví manažéri by museli nadefinovať aj priradenia váh faktorov ku každej otázke a odpovedi. Znamená to, že niektorá otázka alebo odpoveď v dotazníku by mohla mať väčšiu váhu ako nejaká iná. Tieto dotazníky by manažéri priradili pracovníkom podľa ich funkcií teda rozdelenie funkcií pracovníkov by bolo treba ešte viac rozpracovať. Po prihlásení pracovník by bol upozornený na nový dotazník, ktorý by musel bezpodmienečne vyplniť. Po každom vyplnenom dotazníku by bol výsledok presnejší. Systém by automaticky aktualizoval hodnoty v databáze a z týchto hodnôt by postupne vypočítaval závažnosť rizík. Bolo by treba implementovať do systému aj funkciu, ktorá by poskytovala projektovým manažérom prehľad o ich meraniach. Mali by k dispozícii aj štatistiky o vyplnených a nevyplnených dotazníkoch.

Dôležitý smer vývoja vidím aj v detailnejšom rozpracovaní riadenia projektov a zdrojov. Okrem riadenia rizík by systém mohol poskytovať aj viac možností a funkcií na samotné riadenie projektov, procesov a zdrojov. Mohol by zobrazovať jednotlivé fázy vývoja softwaru aj pomocou Ganttovho diagramu, čo je v komerčných aplikáciách riadenia projektov základnou funkciou. V tejto časti by mohlo byť rozpracované aj riadenie zdrojov, ktoré by som spojil s rizikami a tak vytváral štatistiky.

Určite užitočným rozšírením by mohla byť aj funkcia na posielanie správ priamo cez systém. Fungovalo by to ako emailová schránka pracovníkov, kde by dostávali všetky dôležité správy, ktoré sa ich týkajú v rámci systému. Napríklad pri priradení pracovníka k nejakému projektu, alebo pri priradení k riziku ako vlastníka, by systém upozorňoval na tieto skutočnosti pomocou správ. Systém by mohol skontrolovať aj uplynulú dobu od identifikácie rizika. Keby táto hodnota presiahla dovolený limit, systém by upozorňoval vlastníka rizika a projektového manažéra cez správu, že k danému riziku nebola priradená zatiaľ žiadna reakcia.

Do aplikácie ako ďalšie rozsiahle rozšírenie by mohla byť implementovaná aj funkcia na dolovanie frekventovaných množín rizík. Frekventované množiny sú množiny hodnôt, ktoré sa často vyskytujú v danej vzorke dát, v mojom prípade hlavne v rizikách. Tieto množiny by potom poskytovali dôležité štatistiky pre ďalšie skúsenosti riadenia a rozhodovania. Bolo by to vlastne získavanie zaujímavých informácií týkajúcich sa rizík, ktoré nie sú triviálne a ktoré nie je možné vidieť na prvý pohľad.

Určite by sa podobných rozšírení našlo veľa, ale v budúcnosti, keď na tomto projekte budem pokračovať, by som sa pokúsil najprv implementovať vyššie uvedené rozšírenia. Samozrejme môže nastať situácia, že po úspešnom fungovaní systému jednotlivé rozšírenie bude potreba zmeniť. Keďže prípadná implementácia bude postupne prebiehať, bolo by vhodné vytvoriť si aj internetovú stránku na ktorej by bolo možné stiahnuť aktuálnu verziu programu. Táto stránka by obsahovala popisy úprav, nové vlastnosti a aktualizovanú dokumentáciu. Nechýbalo by ani diskusné fórum na riešenie rôznych problémov.

9 Záver

V prvej časti tejto práce bolo popísané, aké je vlastne dôležité projektové riadenie a riadenie rizík v IT projektoch. Úspech alebo neúspech projektu nie je závislé na jedinom rozhodnutí, na jedinom kroku v priebehu projektu, ale závisí od mnohých faktorov. Preto ak spravíme jednu chybu neznamená to, že môžeme celý projekt ukončiť s neúspechom.

Všetky procesy riadenia projektu sú dôležité a nemali by sme vynechávať ani jeden. Riadenie rizík je však činnosť, ktorá je na prvý pohľad v procese riadenia niečím extra, ktorá síce môže pozitívne ovplyvniť celkový výsledok projektu, ale nie je nutná pre úspešné dokončenie projektu. Opak je však pravdou. V praxi sa často stáva, že manažér projektu pokladá riadenie rizík za zbytočné zdržovanie ho od „dôležitejšej“ roboty. A tak práve manažment rizík je v procese riadenia pre mnohé projekty osudným. Vzhľadom k jedinečnej povahe projektu je riziko súčasťou každého z nich. Mnohé organizácie napriek tomu riadenia rizík robia nedostatočne a nesprávne, alebo dokonca to nerobia vôbec.

Po úspešnej analýze týchto problémov som vypracoval špecifikáciu požiadaviek a návrh celého systému. Tento návrh som bral ako základ pri implementácii, avšak pri vývoji vznikli aj nové nápady a nedostatky, ktorými som návrh priebežne doplnil. Tento proces je však prirodzený a nazýva sa životný cyklus vývoja projektov. Systém, ktorý som vytvoril však spĺňa všetky požiadavky, ktoré boli uvedené v zadaní. Keďže neexistujú žiadne všeobecne rozsiahle systémy na riadenie rizík, inšpirácia a porovnávanie s takými komerčnými systémami nebola možná. Celý vývoj grafického používateľského rozhrania a funkcií preto prebiehal len podľa mojich predstáv a pomocou literatúr.

Systém, ktorý som vytvoril však poskytuje aj prehľadné a príjemné viacpoužívateľské rozhranie, pomocou ktorého je možné riadiť riziká v IT jednoducho. Kládol som veľký dôraz na to, aby komunikácia programu s používateľom bola navrhnutá pre potreby používateľa a v žiadnom prípade nie pre potreby programátora. Riziká sú priradené na úroveň procesov a je z nich vytvorená aj centrálna databáza. Systém poskytuje pre projektových manažérov všetky dôležité štatistiky rôzneho charakteru. Umožňuje aj export a import rizík pre prípadné práce s ním pomocou iných aplikácií. Všetky dôležité informácie je možné priamo zo systému vytlačiť cez tlačiareň, alebo vyexportovať do PDF súboru. Okrem funkcií uvedených v zadaní som implementoval aj plus funkcie ako je výpočet očakávanej peňažnej hodnoty, časový graf hodnôt a kvantitatívne, kvalitatívne hodnotenia. Všetky funkcie systému boli dôkladne otestované na používateľoch s rôznymi počítačovými znalosťami.

Táto diplomová práca ma posunula v znalostiach vytváraní informačných systémov o veľký krok dopredu. Získal som znalosti z oblasti .NET technológií a z programovacieho jazyka C#, s ktorými som predtým vôbec nepracoval. Úplne nová, ale veľmi zaujímavá bola aj práca s databázovým serverom Microsoft SQL. Pomocou tejto diplomovej práce som sa zoznámil okrem programátorských znalostí aj s mnohými zaujímavými informáciami z nadväzujúcich oblastí. Sú to predovšetkým oblasti riadenia projektov, procesov a identifikácií rizík. Uvedomil som si, aké je dôležité pri vývoji softwaru samotné riadenie projektov, avšak pre úspešné dokončenie je najdôležitejšie riadenie rizík. V závere tejto práce môžem konštatovať, že manažment rizík v organizáciách je jednoznačným ukazovateľom kvalitného projektového manažmentu.

Cieľom mojej diplomovej práce bolo navrhnúť a implementovať taký softwarový systém, ktorý by pomohol neúspechy v oblasti manažmentu zdokonaľiť. Myslím si, že som cieľ splnil a verím, že tento systém bude v manažmente rizík využívaný.

Literatúra

- [1] Schwalbe, K.: *Řízení projektů v IT*. Computer Press, Brno, 2007. 720s. ISBN 978-80-251-1526-8.
- [2] Rosenau, M. D.: *Řízení projektů*. Academia, Praha, 1998. 344s. ISBN 80-7226-218-1.
- [3] Taylor, J.: *Začínáme řídit projekty*. Computer Press, Brno, 2007. 215s. ISBN 978-80-251-1759-0.
- [4] Kliem, L. R.: *Project Management Metodology*. Marcel Dekker, 1997. 272s. ISBN 082-47-0088-0.
- [5] Schulte, P.: *Complex IT Project Management*. Auerbach Publication, 2004. 314s. ISBN 084-93-1932-3.
- [6] Halanova, Z.: *Manažment projektov: teória vs. prax*. STU FIIT, Bratislava, 2007.
- [7] Doležal, J.: *Projektové řízení a tvorba projektů s podporou v Microsoft Project: Podkladový materiál pro eLearningový kurz*. Brno, 2006.
- [8] Všetěčka, P., Belan, L.: *Projektový manažment I: Vysokoškolská učebnica*. Akadémia Ozbrojených Sí, Liptovský Mikuláš, 2006, ISBN 978-80-804-0298-3.
- [9] Dolanský, V., Mehota, V., Nemec, V.: *Projektový manažment*. Grada, Praha, 1996.
- [10] Smejkal, V., Rais, K.: *Řízení rizik ve firmách a jiných organizacích*. Grada Publishing, Praha, 2006. 30s. ISBN 80-247-1667-4.
- [11] Černá, L., Jakabová, M.: *Project Cycle Managment*. STU MTF, Bratislava, 2007. 101s. ISBN 978-80-227-2822-5.
- [12] Majtán, M.: *Projektový Manažment: Nové trendy v manažmente*. STU MTF, Bratislava, 2002. 117s. ISBN 80-225-1553-1.
- [13] Citizen Kane Blog: *Ruby on Rails III.-MVC* [online]. 2007 [cit. 2009-01-03] Dostupný z WWW: <<http://kane.sk/2007/11/02/ruby-on-rails-iii-mvc/>>.
- [14] Microsoft Corporation: *Microsoft Developer Network : C# Programming Guide* [online]. 2008 [cit. 2009-05-06]. Dostupný z WWW: <[http://msdn.microsoft.com/en-us/library/67ef8sbd\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/67ef8sbd(VS.80).aspx)>.
- [15] Jakel, M.: *Sblížení s Microsoft SQL Server* [online]. 2002 [cit. 2009-05-06]. Dostupný z WWW: <<http://interval.cz/clanky/sblizeni-s-microsoft-sql-server/>>.
- [16] Troelsen, A.: *C# a .NET 2.0 profesionálně*. Zoner Press, Brno, 2006. 1197s. ISBN 80-86815-42-0.
- [17] Sourceforge.NET: *Open Source Softwares: iTextSharp* [online]. 2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://itextsharp.sourceforge.net/>>.

Zoznam príloh

Príloha 1: CD, obsahujúce zdrojové kódy, skompilovanú aplikáciu, databázový skript a dokumentáciu vo formáte pdf a doc.